

# **EXHIBIT 5**



## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
**United States Patent and Trademark Office**  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
13/849,606	9218156	2183	9200

**Correspondence Address/Fee Address Change**

The following fields have been set to Customer Number 144359 on 07/31/2017

- Correspondence Address
- Maintenance Fee Address

The address of record for Customer Number 144359 is:

144359  
 Blueshift IP LLC  
 1 Broadway, 14th Floor  
 Cambridge, MA 02142



UNITED STATES PATENT AND TRADEMARK OFFICE  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 9,218,156 B2  
APPLICATION NO. : 13/849606  
DATED : December 22, 2015  
INVENTOR(S) : Bates

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In Column 9, Lines 30-31, delete "I/O controller 108," and insert -- I/O unit 108, --, therefor.

In Column 20, Line 1, delete "&" and insert -- % --, therefor.

In Column 21, Line 3, delete "0(M×M)" and insert -- O(M×M) --, therefor.

In Column 21, Line 5, delete "0(M×M)" and insert -- O(M×M) --, therefor.

In Column 22, Line 6, delete "Kahan (Kahan," and insert -- Kahan --, therefor.

In Column 30, Line 7, in Claim 2, delete "The method" and insert -- The device --, therefor.

Signed and Sealed this  
Twenty-third Day of May, 2017



Michelle K. Lee  
*Director of the United States Patent and Trademark Office*

ATTORNEY'S DOCKET NO: A0006-1001C2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent No.: US9218156B2  
Serial No.: 13/849,606  
Confirmation No.: 6059  
Title: Processing with compact arithmetic  
processing element  
First Inventor: Bates, Joseph  
Filing Date: March 25, 2013  
Examiner: Yaary; Michael D  
Art Unit: 2193

**STATEMENT ACCOMPANYING REQUEST FOR CERTIFICATE OF CORRECTION**

Please correct the above-referenced patent in accordance with the attached draft Certificate of Correction.

At least some errors were the fault of the Applicant and, accordingly, please charge \$100.00 to our Deposit Account No. 50-1797. In the event that a further fee is required, please charge the amount to the same Deposit Account.

The exact locations where the errors appear in the patent and patent application are as follows:

In Column 9, Lines 30-31, delete "I/O controller 108," and insert - - I/O unit 108, - -, therefor.  
(SUBSTITUTE SPECIFICATION DATED JULY 30, 2013, PAGE 16, PARAGRAPH [0049], LINES 7-8)

In Column 20, Line 1, delete "&" and insert - - % - -, therefor.  
(SUBSTITUTE SPECIFICATION DATED JULY 30, 2013, PAGE 34, PARAGRAPH [0107], LINE 3)

In Column 21, Line 3, delete "O(M×M)" and insert - - O(M×M) - -, therefor.  
(SUBSTITUTE SPECIFICATION DATED JULY 30, 2013, PAGE 36, PARAGRAPH [0112], LINE 4)

In Column 21, Line 5, delete "O(M×M)" and insert - - O(M×M) - -, therefor.  
(SUBSTITUTE SPECIFICATION DATED JULY 30, 2013, PAGE 36, PARAGRAPH [0112], LINE 6)

In Column 22, Line 6, delete "Kahan (Kahan," and insert - - Kahan - -, therefor.  
(SUBSTITUTE SPECIFICATION DATED JULY 30, 2013, PAGE 37, PARAGRAPH [0118], LINE 7)

In Column 30, Line 7, in Claim 2, delete "The method" and insert - - The device - -, therefor.  
(ORIGINALLY FILED SPECIFICATION, IN THE CLAIMS DATED JULY 30, 2013, PAGE 51, CLAIM 2, LINE 1)

The requested corrections are attached on Form PTO 1050.

Application Serial No. 13/849,606

Attorney Docket No. A0006-1001C2

**CONCLUSIONS**

If this response is not considered timely filed and if a request for extension of time is otherwise absent, applicant hereby requests any extension of time. Please charge any fees or make any credits, to Deposit Account No. 50/1797.

Respectfully submitted,

/Robert Plotkin/

Robert Plotkin, Esq.

Reg. No. 43,861

January 19, 2016

Date

Robert Plotkin, P.C.

15 New England Executive Park

Burlington, MA 01803 USA

Tel: (978) 318-9914

Fax: (978) 318-9060

## UNITED STATES PATENT AND TRADEMARK OFFICE CERTIFICATE OF CORRECTION

Page 1 of 1

PATENT NO. : 9,218,156 B2  
 APPLICATION NO. : 13/849,606  
 ISSUE DATE : December 22, 2015  
 INVENTOR(S) : Bates

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In Column 9, Lines 30-31, delete "I/O controller 108," and insert - - I/O unit 108, - -, therefor.

In Column 20, Line 1, delete "&" and insert - - % - -, therefor.

In Column 21, Line 3, delete "0(M×M)" and insert - - O(M×M) - -, therefor.

In Column 21, Line 5, delete "0(M×M)" and insert - - O(M×M) - -, therefor.

In Column 22, Line 6, delete "Kahan (Kahan," and insert - - Kahan - -, therefor.

In Column 30, Line 7, in Claim 2, delete "The method" and insert - - The device - -, therefor.

### MAILING ADDRESS OF SENDER (Please do not use customer number below):

Robert Plotkin, P.C.  
 15 New England Executive Park,  
 Burlington, MA 01803, USA

This collection of information is required by 37 CFR 1.322, 1.323, and 1.324. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1.0 hour to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Attention Certificate of Corrections Branch, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

*If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

Electronic Patent Application Fee Transmittal				
<b>Application Number:</b>		13849606		
<b>Filing Date:</b>		25-Mar-2013		
<b>Title of Invention:</b>		PROCESSING WITH COMPACT ARITHMETIC PROCESSING ELEMENT		
<b>First Named Inventor/Applicant Name:</b>		Joseph Bates		
<b>Filer:</b>		Robert Plotkin/Karen Del Greco		
<b>Attorney Docket Number:</b>		A0006-1001C2		
Filed as Small Entity				
<b>Filing Fees for Utility under 35 USC 111(a)</b>				
<b>Description</b>	<b>Fee Code</b>	<b>Quantity</b>	<b>Amount</b>	<b>Sub-Total in USD(\$)</b>
<b>Basic Filing:</b>				
<b>Pages:</b>				
<b>Claims:</b>				
<b>Miscellaneous-Filing:</b>				
<b>Petition:</b>				
<b>Patent-Appeals-and-Interference:</b>				
<b>Post-Allowance-and-Post-Issuance:</b>				
Certificate of Correction	2811	1	100	100

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				100

**Electronic Acknowledgement Receipt**

<b>EFS ID:</b>	24656573
<b>Application Number:</b>	13849606
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	6059
<b>Title of Invention:</b>	PROCESSING WITH COMPACT ARITHMETIC PROCESSING ELEMENT
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates
<b>Customer Number:</b>	24208
<b>Filer:</b>	Robert Plotkin/Karen Del Greco
<b>Filer Authorized By:</b>	Robert Plotkin
<b>Attorney Docket Number:</b>	A0006-1001C2
<b>Receipt Date:</b>	19-JAN-2016
<b>Filing Date:</b>	25-MAR-2013
<b>Time Stamp:</b>	14:11:39
<b>Application Type:</b>	Utility under 35 USC 111(a)

**Payment information:**

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$ 100
RAM confirmation Number	426
Deposit Account	501797
Authorized User	DEL GRECO, KAREN

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 CFR 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 CFR 1.17 (Patent application and reexamination processing fees)



Charge any Additional Fees required under 37 CFR 1.19 (Document supply fees)

Charge any Additional Fees required under 37 CFR 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 CFR 1.21 (Miscellaneous fees and charges)

## File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Certificate of Correction	RequestForCoC_A0006-1001C2.pdf	56424	no	3
			049ef4ca2b1e0b227bb5aa4c1e9ef92b403251c3		
Warnings:					
Information:					
2	Request for Certificate of Correction	COC_A0006-1001C2.pdf	102194	no	1
			2034350b99d00274fe312866494f76bc2e1be9c7		
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	30526	no	2
			1afcb3eec4bfc556c19b9059e73a14fb5282b54e		
Warnings:					
Information:					
Total Files Size (in bytes):			189144		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

### New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

### National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

### New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.



## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
**United States Patent and Trademark Office**  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/849,606	12/22/2015	9218156	A0006-1001C2	6059

24208 7590 12/02/2015

Robert Plotkin, P.C.  
 15 New England Executive Park  
 Burlington, MA 01803

**ISSUE NOTIFICATION**

The projected patent number and issue date are specified above.

**Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)**  
 (application filed on or after May 29, 2000)

The Patent Term Adjustment is 158 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (<http://pair.uspto.gov>).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site <http://pair.uspto.gov> for additional applicants):

Singular Computing LLC, Newton, MA;  
 Joseph Bates, Newton, MA;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit [SelectUSA.gov](http://SelectUSA.gov).

Document Description: Issue Fee Payment (PTO-85B)

**Issue Fee Transmittal Form**

Application Number	Filing Date	First Named Inventor	Atty. Docket No.	Confirmation No.
13849606	25-Mar-2013	Joseph Bates	A0006-1001C2	6059

**TITLE OF INVENTION :**

PROCESSING WITH COMPACT ARITHMETIC PROCESSING ELEMENT

Entity Status	Application Type	Art Unit	Class - Subclass	EXAMINER
Small	Utility under 35 USC 111(a)	2183	209000	MICHAEL YAARY
Issue Fee Due	Publication Due	Total Fee(s) Due	Date Due	Prev. Paid Fee
\$480	\$0	\$480	18-Nov-2015	\$0

**1.Change of Correspondence Address and/or Indication Of Fee Address (37 CFR 1.33 & 1.363)**

Current Correspondence Address:	Current Indicated Fee Address :
24208 Robert Plotkin, P.C.  15 New England Executive Park  Burlington MA 01803 UNITED STATES 978-318-9914 RPLOTKIN@RPLOTKIN.COM	
<input type="checkbox"/> Change of correspondence address requested, system generated AIA/122-EFS form attached	<input type="checkbox"/> Fee Address indication requested, system generated SB/47-EFS form attached

**2.Entity Status****Change in Entity Status**

Applicant certifying micro entity status; system generated Micro Entity certification form attached. See 37 CFR 1.29.

Note: Absent a valid certification of micro entity status, issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment. If this box is checked, you will be prompted to choose a micro entity status on the gross income basis (37 CFR 1.29(a)) or the institution of higher education basis (37 CFR 1.29(d)), and make the applicable certification online.



Applicant asserting small entity status. See 37 CFR 1.27.

Note: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.



Applicant changing to regular undiscounted fee status.

Note: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.



Document Description: Issue Fee Payment (PTO-85B)

**3.The Following Fee(s) Are Submitted:**☒ Issue Fee☐ I authorize USPTO to apply my previously paid issue fee to the current fees due☐ Publication Fee☐ The Director is hereby authorized to apply my previously paid issue fee to the current fee due and to charge deficient fees to Deposit Account Number \_\_\_\_\_☐ Advance Order - # of copies \_\_\_\_\_

If in **addition** to the payment of the issue fee amount submitted with this form, there are any discrepancies in any amount(s) due, the Director is authorized to charge any deficiency, or credit any overpayment, to Deposit Account Number 501797.

☒ The **issue fee must be submitted** with this form. **If payment of the issue fee does not accompany this form, checking this box and providing a deposit account number will NOT be effective to satisfy full payment of the fee(s) due.**

**4.Firm and/or Attorney Names To Be Printed****NOTE: If no name is listed, no name will be printed**

For printing on the patent front page, list to be displayed as entered

1. ROBERT PLOTKIN, P.C.

2. ROBERT PLOTKIN

3.

**5.Assignee Name(s) and Residence Data To Be Printed**

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

Name	City	State	Country	Category
Singular Computing LLC	Newton	MASSACHUSETTS	united states	corporation

**6.Signature**

I certify, in accordance with 37 CFR 1.4(d)(4) that I am an attorney or agent registered to practice before the Patent and Trademark Office who has filed and has been granted power of attorney in this application. I also certify that this Fee(s) Transmittal form is being transmitted to the USPTO via EFS-WEB on the date indicated below.

<b>Signature</b>	/Robert Plotkin/	<b>Date</b>	11-16-2015
<b>Name</b>	Robert Plotkin	<b>Registration Number</b>	43861

Electronic Patent Application Fee Transmittal				
<b>Application Number:</b>		13849606		
<b>Filing Date:</b>		25-Mar-2013		
<b>Title of Invention:</b>		PROCESSING WITH COMPACT ARITHMETIC PROCESSING ELEMENT		
<b>First Named Inventor/Applicant Name:</b>		Joseph Bates		
<b>Filer:</b>		Robert Plotkin		
<b>Attorney Docket Number:</b>		A0006-1001C2		
Filed as Small Entity				
<b>Filing Fees for Utility under 35 USC 111(a)</b>				
<b>Description</b>	<b>Fee Code</b>	<b>Quantity</b>	<b>Amount</b>	<b>Sub-Total in USD(\$)</b>
<b>Basic Filing:</b>				
Utility Appl Issue Fee	2501	1	480	480
Publ. Fee- Early, Voluntary, or Normal	1504	1	0	0
<b>Pages:</b>				
<b>Claims:</b>				
<b>Miscellaneous-Filing:</b>				
<b>Petition:</b>				
<b>Patent-Appeals-and-Interference:</b>				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				480

**Electronic Acknowledgement Receipt**

<b>EFS ID:</b>	24086968
<b>Application Number:</b>	13849606
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	6059
<b>Title of Invention:</b>	PROCESSING WITH COMPACT ARITHMETIC PROCESSING ELEMENT
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates
<b>Customer Number:</b>	24208
<b>Filer:</b>	Robert Plotkin
<b>Filer Authorized By:</b>	
<b>Attorney Docket Number:</b>	A0006-1001C2
<b>Receipt Date:</b>	16-NOV-2015
<b>Filing Date:</b>	25-MAR-2013
<b>Time Stamp:</b>	10:34:57
<b>Application Type:</b>	Utility under 35 USC 111(a)

**Payment information:**

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	\$480
RAM confirmation Number	9593
Deposit Account	501797
Authorized User	PLOTKIN, ROBERT

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

### File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	Web85b.pdf	46128	no	2
			0f453d8c32fb1fbc613a13fddda8ac20d68978a		

### Warnings:

### Information:

2	Fee Worksheet (SB06)	fee-info.pdf	32159	no	2
			e3def0c042dfc4a058a97446c2b0a70090586001		

### Warnings:

### Information:

<b>Total Files Size (in bytes):</b>	78287
-------------------------------------	-------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

#### New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

#### National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

#### New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.





## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
 United States Patent and Trademark Office  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

## NOTICE OF ALLOWANCE AND FEE(S) DUE

24208 7590 08/18/2015  
 Robert Plotkin, P.C.  
 15 New England Executive Park  
 Burlington, MA 01803

EXAMINER

YAARY, MICHAEL D

ART UNIT

PAPER NUMBER

2193

DATE MAILED: 08/18/2015

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

13/849,606

03/25/2013

Joseph Bates

A0006-1001C2

6059

TITLE OF INVENTION: PROCESSING WITH COMPACT ARITHMETIC PROCESSING ELEMENT

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$480	\$0	\$0	\$480	11/18/2015

**THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.**

**THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.**

## HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

**IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.**

## PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: **Mail** **Mail Stop ISSUE FEE**  
**Commissioner for Patents**  
**P.O. Box 1450**  
**Alexandria, Virginia 22313-1450**  
**or Fax (571)-273-2885**

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

24208 7590 08/18/2015  
**Robert Plotkin, P.C.**  
**15 New England Executive Park**  
**Burlington, MA 01803**

## Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/849,606	03/25/2013	Joseph Bates	A0006-1001C2	6059

TITLE OF INVENTION: PROCESSING WITH COMPACT ARITHMETIC PROCESSING ELEMENT

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$480	\$0	\$0	\$480	11/18/2015

EXAMINER	ART UNIT	CLASS-SUBCLASS
YAARY, MICHAEL D	2193	708-209000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

- (1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
- (2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 \_\_\_\_\_  
 2 \_\_\_\_\_  
 3 \_\_\_\_\_

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent): ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:

- ☐ Issue Fee
- ☐ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies \_\_\_\_\_

4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above)

- ☐ A check is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☐ The director is hereby authorized to charge the required fee(s), any deficiency, or credits any overpayment, to Deposit Account Number \_\_\_\_\_ (enclose an extra copy of this form).

5. Change in Entity Status (from status indicated above)

- ☐ Applicant certifying micro entity status. See 37 CFR 1.29
- ☐ Applicant asserting small entity status. See 37 CFR 1.27
- ☐ Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature \_\_\_\_\_

Date \_\_\_\_\_

Typed or printed name \_\_\_\_\_

Registration No. \_\_\_\_\_



## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
 United States Patent and Trademark Office  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/849,606	03/25/2013	Joseph Bates	A0006-1001C2	6059

24208	7590	08/18/2015
-------	------	------------

Robert Plotkin, P.C.  
 15 New England Executive Park  
 Burlington, MA 01803

EXAMINER
YAARY, MICHAEL D

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 08/18/2015

**Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)**

(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

**OMB Clearance and PRA Burden Statement for PTOL-85 Part B**

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**Privacy Act Statement**

**The Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.



<b>Notice of Allowability</b>	<b>Application No.</b> 13/849,606	<b>Applicant(s)</b> BATES, JOSEPH	
	<b>Examiner</b> MICHAEL D. YAARY	<b>Art Unit</b> 2193	<b>AIA (First Inventor to File) Status</b> No

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--**

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 07/24/2015.  
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on \_\_\_\_.
2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on \_\_\_\_; the restriction requirement and election have been incorporated into this action.
3. ☒ The allowed claim(s) is/are 1-42. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see [http://www.uspto.gov/patents/init\\_events/pph/index.jsp](http://www.uspto.gov/patents/init_events/pph/index.jsp) or send an inquiry to [PPHfeedback@uspto.gov](mailto:PPHfeedback@uspto.gov).
4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

**Certified copies:**

a) ☐ All    b) ☐ Some    \*c) ☐ None of the:

1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

\* Certified copies not received: \_\_\_\_.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.  
**THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.**

5. ☐ CORRECTED DRAWINGS ( as "replacement sheets") must be submitted.  
☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date \_\_\_\_.

**Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).**

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

**Attachment(s)**

<ol style="list-style-type: none"> <li>1. <input type="checkbox"/> Notice of References Cited (PTO-892)</li> <li>2. <input checked="" type="checkbox"/> Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date <u>06/25/2015</u></li> <li>3. <input type="checkbox"/> Examiner's Comment Regarding Requirement for Deposit of Biological Material</li> <li>4. <input type="checkbox"/> Interview Summary (PTO-413), Paper No./Mail Date ____.</li> </ol>	<ol style="list-style-type: none"> <li>5. <input type="checkbox"/> Examiner's Amendment/Comment</li> <li>6. <input checked="" type="checkbox"/> Examiner's Statement of Reasons for Allowance</li> <li>7. <input type="checkbox"/> Other ____.</li> </ol>
--	---

/MICHAEL D YAARY/ Primary Examiner, Art Unit 2193	
--	--

Application/Control Number: 13/849,606  
Art Unit: 2193

Page 2

### **REASONS FOR ALLOWANCE**

The following is an examiner's statement of reasons for allowance:

The prior art of record does not explicitly teach or suggest "wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/65,000 through 65,000 and for at least  $X=5\%$  of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least  $X\%$  of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least  $Y=.05\%$  from the result of an exact mathematical calculation of the first operation on the numerical values of that same input' and similarly in independent claims 9, 16, 26, and 33.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Application/Control Number: 13/849,606  
Art Unit: 2193

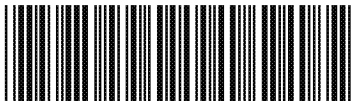
Page 3

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MICHAEL D. YAARY whose telephone number is (571)270-1249. The examiner can normally be reached on Mon-Fri 9 a.m.-5:30 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chat Do can be reached on 571-272-3721. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/MICHAEL D YAARY/  
Primary Examiner, Art Unit 2193

<b>Search Notes</b> 	<b>Application/Control No.</b> 13849606	<b>Applicant(s)/Patent Under Reexamination</b> BATES, JOSEPH
	<b>Examiner</b> MICHAEL D YAARY	<b>Art Unit</b> 2193

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

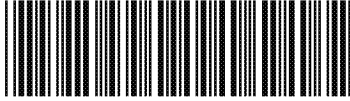
US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES		
Search Notes	Date	Examiner
EAST search with keywords	01/20/2015	MY
NPL search	01/20/2015	MY
CPC search	01/20/2015	MY
Inventor name search	01/20/2015	MY
Updated search	08/12/2015	MY

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner
G06F	7/38	08/12/2015	MY

--	--



<b>Issue Classification</b> 	<b>Application/Control No.</b> 13849606	<b>Applicant(s)/Patent Under Reexamination</b> BATES, JOSEPH
	<b>Examiner</b> MICHAEL D YAARY	<b>Art Unit</b> 2193

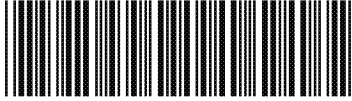
CPC						
Symbol					Type	Version
G06F		7	/	38	F	2013-01-01
G06F		7	/	4833	I	2013-01-01
G06F		7	/	5235	I	2013-01-01
H03K		19	/	17728	I	2013-01-01
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			
			/			

CPC Combination Sets				
Symbol	Type	Set	Ranking	Version

NONE		<b>Total Claims Allowed:</b>	
		42	
(Assistant Examiner)	(Date)		
/MICHAEL D YAARY/ Primary Examiner.Art Unit 2193	08/12/2015	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	6

[illegible]

NONE		Total Claims Allowed:	
(Assistant Examiner)	(Date)	42	
/MICHAEL D YAARY/ Primary Examiner.Art Unit 2193	08/12/2015	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	6

<b>Issue Classification</b> 	<b>Application/Control No.</b> 13849606	<b>Applicant(s)/Patent Under Reexamination</b> BATES, JOSEPH
	<b>Examiner</b> MICHAEL D YAARY	<b>Art Unit</b> 2193

<input checked="" type="checkbox"/> <b>Claims renumbered in the same order as presented by applicant</b> <input type="checkbox"/> <b>CPA</b> <input checked="" type="checkbox"/> <b>T.D.</b> <input type="checkbox"/> <b>R.1.47</b>															
Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1	17	17	33	33										
2	2	18	18	34	34										
3	3	19	19	35	35										
4	4	20	20	36	36										
5	5	21	21	37	37										
6	6	22	22	38	38										
7	7	23	23	39	39										
8	8	24	24	40	40										
9	9	25	25	41	41										
10	10	26	26	42	42										
11	11	27	27												
12	12	28	28												
13	13	29	29												
14	14	30	30												
15	15	31	31												
16	16	32	32												

NONE		<b>Total Claims Allowed:</b>	
(Assistant Examiner)		42	
(Date)			
/MICHAEL D YAARY/ Primary Examiner.Art Unit 2193		08/12/2015	
(Primary Examiner)		(Date)	
		O.G. Print Claim(s)	O.G. Print Figure
		1	6

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application Number	13/849,606
	Filing Date	3/25/2013
	First Named Inventor	Bates
	Art Unit	2193
	Client Number	A0006
Page 1 of 2	Matter Number	1001C2

FOREIGN PATENT DOCUMENTS					
Initials	Cite No.	Document Number	Publication Date	Country	T
/M.Y./	1	2008158822	07/10/2008	JP	
/M.Y./	2	2012530966	12/06/2012	JP	
/M.Y./	3	H0314128	01/22/1991	JP	

OTHER DOCUMENTS					T
Initials	Cite No.	Author, Title, Date, Pages, etc.			
/M.Y./	4	Naohito Nakasato, et al., "A Compiler for High Performance Adaptive Precision Computing," A paper of SACSIS (Symposium on Advanced Computing Systems and Infrastructures) 2008, Information Processing Society of Japan, June 4, 2008, No.5, pp. 149-156.			

Examiner Signature	/Michael Yaary/	Date Considered	08/12/2015
Examiner: Please initial if citation considered, whether or not citation is in conformance with MPEP Section 609. Please draw a line through the citation if it is not in conformance and it is not considered. Please include a copy of this form with the next communication to the applicant.			

<b>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</b>	<b>Application Number</b>	13/849,606
	<b>Filing Date</b>	3/25/2013
	<b>First Named Inventor</b>	Bates
	<b>Art Unit</b>	2193
	<b>Client Number</b>	A0006
Page 2 of 2	<b>Matter Number</b>	1001C2

<b>GENERAL</b>
<p>Pursuant to 37 C.F.R. 1.97 and 1.98 and to the duty of disclosure set forth in 37 C.F.R. 1.56, the Examiner in charge of the above-identified application is requested to consider and make of record the references listed herewith. A copy of each listed reference, other than U.S. patents/applications and references cited in a parent application, is enclosed.</p> <p>Although the information submitted herewith may be "material" to the Examiner's consideration of the subject application, this submission is not intended to constitute an admission that such information is "prior art" as to the claimed invention.</p> <p>In accordance with 37 C.F.R. 1.97(g), the filing of this Information Disclosure Statement shall not be construed to mean that a search has been made.</p>

<b>TIMING</b>
<p>In accordance with 37 CFR 1.97(c), this Information Disclosure Statement is being filed after the period specified in 37 CFR 1.97(b) and before the mailing date of a Final Action under 37 CFR 1.113, a Notice of Allowance under 37 CFR 1.311, or an Action that otherwise closes prosecution in the application.</p>

<b>CERTIFICATION STATEMENT</b>
<p>Each item of information contained in this information disclosure statement was first cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.</p>

<b>FEE</b>
<p>No fee is required. This Information Disclosure Statement is being filed in accordance with 37 CFR 1.97(c), and is accompanied by the certification statement of 37 CFR 1.97(e)(1).</p> <p>If necessary, the Director is hereby authorized to charge or credit Deposit Account No. 50-1797 for any additional fees, or any underpayment or credit for overpayment in connection herewith. Please reference attorney docket number A0006-1001C2 for any such charge or credit.</p>

<b>SIGNATURE</b>			
Signature	/Robert Plotkin/	Date	June 25, 2015
Name	Robert Plotkin	Registration Number	43861

## EAST Search History

## EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	85	(low near5 precision) near10 (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2015/08/12 10:36
L2	863	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution) and mean	US-PGPUB; USPAT	OR	ON	2015/08/12 10:36
L5	130	G06F7/\$.cpc. and range and wide and mean and fpga	US-PGPUB; USPAT	OR	ON	2015/08/12 10:37
L6	84	G06F7/\$.cpc. and range and wide and mean and fpga and (multiply\$3 or multiplication)	US-PGPUB; USPAT	OR	ON	2015/08/12 10:37
L7	1	("20140095571").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2015/08/12 10:41
S2	20	((("5887160") or ("5867683") or ("5809320") or ("5293500") or ("5226166") or ("4380046") or ("5170484") or ("5801715") or ("5966528") or ("6173388") or ("6405185") or ("6728871") or ("6859869") or ("7243333") or ("20050076187") or ("20060031659") or ("20070124565") or ("20070150698") or ("20070247189") or ("20080059764"))).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/01/24 13:48
S3	20	((("5887160") or ("5867683") or ("5809320") or ("5293500") or ("5226166") or ("4380046") or ("5170484") or ("5801715") or ("5966528") or ("6173388") or ("6405185") or ("6728871") or ("6859869") or ("7243333") or ("20050076187") or ("20060031659") or ("20070124565") or ("20070150698") or ("20070247189") or ("20080059764"))).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/01/24 15:27
S4	13	S3 and (width or wide)	US-PGPUB; USPAT	OR	ON	2012/01/24 15:27
S5	20	((("5887160") or ("5867683") or ("5809320") or ("5293500") or ("5226166") or ("4380046") or ("5170484") or ("5801715") or ("5966528") or ("6173388") or ("6405185") or ("6728871") or ("6859869") or ("7243333") or ("20050076187") or ("20060031659") or ("20070124565") or ("20070150698") or ("20070247189") or ("20080059764"))).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/02/06 10:38
S6	6	S5 and precision	US-PGPUB; USPAT	OR	ON	2012/02/06 10:42



S7	1	("20100325186").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/02/06 10:46
S8	13	S5 and (width or wide)	US-PGPUB; USPAT	OR	ON	2012/02/06 11:02
S9	6	S8 and precision	US-PGPUB; USPAT	OR	ON	2012/02/06 11:02
S10	22628	(low near5 precision)	US-PGPUB; USPAT	OR	ON	2012/02/06 14:10
S11	632	(low near5 precision) and (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2012/02/06 14:11
S12	59	(low near5 precision) near10 (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2012/02/06 14:11
S13	46	(low near5 precision) near10 (high near5 dynamic) and signal\$1	US-PGPUB; USPAT	OR	ON	2012/02/06 14:11
S14	4323	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution)	US-PGPUB; USPAT	OR	ON	2012/02/06 15:04
S15	611	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution) and mean	US-PGPUB; USPAT	OR	ON	2012/02/06 15:05
S16	140	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution) and mean and precision	US-PGPUB; USPAT	OR	ON	2012/02/06 15:05
S17	28	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution) and mean and ((high or low) near5 precision)	US-PGPUB; USPAT	OR	ON	2012/02/06 15:05
S18	1	("20080276232").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/02/06 15:31
S19	1	("7549145").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/02/06 15:34
S20	682	382/255.ccls.	US-PGPUB; USPAT	OR	ON	2012/02/07 10:34
S21	494	382/255.ccls. and range	US-PGPUB; USPAT	OR	ON	2012/02/07 10:35
S22	208	382/255.ccls. and range and wide	US-PGPUB; USPAT	OR	ON	2012/02/07 10:35
S23	86	382/255.ccls. and range and wide and mean	US-PGPUB; USPAT	OR	ON	2012/02/07 10:35
S24	42	382/255.ccls. and range and wide and mean and blur	US-PGPUB; USPAT	OR	ON	2012/02/07 10:35
S25	92	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm. and mean	US-PGPUB; USPAT	OR	ON	2012/02/07 10:37
S26	1	("20080276232").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/02/07 10:50
S35	1	("8150902").PN.	USPAT; USOCR	OR	OFF	2012/11/14 09:41
S36	696	(low near5 precision) and (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2012/11/15 10:38
S37	63	(low near5 precision) near10 (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2012/11/15 10:39
S38	2	(low near5 precision) near10 (high near5 dynamic) and fpga	US-PGPUB; USPAT	OR	ON	2012/11/15 10:39

S39	2	(low near5 precision) near10 (high near5 dynamic) and "708".clas.	US-PGPUB; USPAT	OR	ON	2012/11/15 10:39
S40	20	((("5887160") or ("5867683") or ("5809320") or ("5293500") or ("5226166") or ("4380046") or ("5170484") or ("5801715") or ("5966528") or ("6173388") or ("6405185") or ("6728871") or ("6859869") or ("7243333") or ("20050076187") or ("20060031659") or ("20070124565") or ("20070150698") or ("20070247189") or ("20080059764"))).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/11/15 10:40
S43	1	("8407273").PN.	USPAT; USOCR	OR	OFF	2015/01/20 10:40
S44	892	(low near5 precision) and (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2015/01/20 12:12
S45	79	(low near5 precision) near10 (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2015/01/20 12:12
S46	79	(low near5 precision) near10 (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2015/01/20 12:13
S47	18	G06F7/\$.cpc. and (low near5 precision) near10 (high near5 dynamic)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 12:24
S48	73	G06F7/\$.cpc. and range and wide and mean and blur	US-PGPUB; USPAT	OR	ON	2015/01/20 13:02

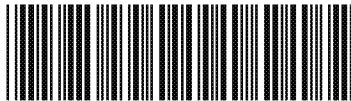
**EAST Search History (Interference)**

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L3	2	G06F7/\$.cpc. and (low near5 precision).clm. near10 (high near5 dynamic)	USPAT; UPAD	OR	ON	2015/08/12 10:36
L4	2	G06F7/\$.cpc. and (low near5 precision).clm. near10 (high near5 dynamic).clm.	USPAT; UPAD	OR	ON	2015/08/12 10:36
S27	60	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm. and mean	USPAT; UPAD	OR	ON	2012/02/07 10:37
S28	8	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm. and mean.clm.	USPAT; UPAD	OR	ON	2012/02/07 10:37
S29	0	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm. and mean.clm. and precision.clm.	USPAT; UPAD	OR	ON	2012/02/07 10:38
S30	6	("708".clas. or "712".clas. or "382.clas") and mean.clm. and precision.clm.	USPAT; UPAD	OR	ON	2012/02/07 10:38
S31	17	("708".clas. or "712".clas. or "382.clas") and (low near3 precision).clm.	USPAT; UPAD	OR	ON	2012/02/07 10:39
S32	0	("708".clas. or "712".clas. or "382.clas") and (low near3 precision).clm. and deblur.clm.	USPAT; UPAD	OR	ON	2012/02/07 10:39
S33	12	("708".clas. or "712".clas. or "382.clas") and (low near3 precision).clm. and (high near3 precision)	USPAT; UPAD	OR	ON	2012/02/07 10:40
S34	8	("708".clas. or "712".clas. or "382.clas") and (low near3 precision).clm. and (high near3	USPAT; UPAD	OR	ON	2012/02/07 10:40



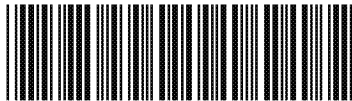
		precision).clm.				
S41	528	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm.	USPAT; UPAD	OR	ON	2012/11/15 10:43
S42	528	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm.	USPAT; UPAD	OR	ON	2012/11/15 10:43

8/ 12/ 2015 11:27:31 AM  
C:\ Users\ myaary\ Documents\ EAST\ Workspaces\ 12816201.wsp

<b><i>Index of Claims</i></b>  	<b>Application/Control No.</b>  13849606	<b>Applicant(s)/Patent Under Reexamination</b>  BATES, JOSEPH
	<b>Examiner</b>  MICHAEL D YAARY	<b>Art Unit</b>  2193

✓	<b>Rejected</b>	-	<b>Cancelled</b>	N	<b>Non-Elected</b>	A	<b>Appeal</b>
=	<b>Allowed</b>	÷	<b>Restricted</b>	I	<b>Interference</b>	O	<b>Objected</b>

<input checked="" type="checkbox"/> Claims renumbered in the same order as presented by applicant				<input type="checkbox"/> CPA				<input checked="" type="checkbox"/> T.D.				<input type="checkbox"/> R.1.47			
CLAIM		DATE													
Final	Original	01/20/2015	08/12/2015												
1	1	✓	=												
2	2	O	=												
3	3	O	=												
4	4	O	=												
5	5	O	=												
6	6	O	=												
7	7	O	=												
8	8	O	=												
9	9	=	=												
10	10	=	=												
11	11	=	=												
12	12	=	=												
13	13	=	=												
14	14	=	=												
15	15	=	=												
16	16	✓	=												
17	17	✓	=												
18	18	✓	=												
19	19	O	=												
20	20	O	=												
21	21	O	=												
22	22	O	=												
23	23	O	=												
24	24	O	=												
25	25	O	=												
26	26	=	=												
27	27	=	=												
28	28	=	=												
29	29	=	=												
30	30	=	=												
31	31	=	=												
32	32	=	=												
33	33	✓	=												
34	34	O	=												
35	35	O	=												
36	36	O	=												

<b><i>Index of Claims</i></b>  	<b>Application/Control No.</b>  13849606	<b>Applicant(s)/Patent Under Reexamination</b>  BATES, JOSEPH
	<b>Examiner</b>  MICHAEL D YAARY	<b>Art Unit</b>  2193

✓	<b>Rejected</b>	-	<b>Cancelled</b>	N	<b>Non-Elected</b>	A	<b>Appeal</b>
=	<b>Allowed</b>	÷	<b>Restricted</b>	I	<b>Interference</b>	O	<b>Objected</b>

<input checked="" type="checkbox"/> Claims renumbered in the same order as presented by applicant				<input type="checkbox"/> CPA				<input checked="" type="checkbox"/> T.D.				<input type="checkbox"/> R.1.47			
CLAIM		DATE													
Final	Original	01/20/2015	08/12/2015												
37	37	○	=												
38	38	○	=												
39	39	○	=												
40	40	○	=												
41	41	○	=												
42	42	○	=												

Doc Code: DIST.E.FILE

Document Description: Electronic Terminal Disclaimer - Filed

U.S. Patent and Trademark Office  
Department of Commerce

Electronic Petition Request	<b>TERMINAL DISCLAIMER TO OBIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT</b>	
Application Number	13849606	
Filing Date	25-Mar-2013	
First Named Inventor	Joseph Bates	
Attorney Docket Number	A0006-1001C2	
Title of Invention	Processing with Compact Arithmetic Processing Element	
<input checked="" type="checkbox"/> Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action  <input checked="" type="checkbox"/> This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.		
Owner	Percent Interest	
Singular Computing LLC	100%	
<p>The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)</p> <p>8407273</p> <p>as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.</p> <p>In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:</p> <ul style="list-style-type: none"> <li>- expires for failure to pay a maintenance fee;</li> <li>- is held unenforceable;</li> <li>- is found invalid by a court of competent jurisdiction;</li> <li>- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;</li> <li>- has all claims canceled by a reexamination certificate;</li> <li>- is reissued; or</li> <li>- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.</li> </ul> <p><input checked="" type="radio"/> Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.</p>		

- ☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.

Applicant claims the following fee status:

- ☒ Small Entity
- ☐ Micro Entity
- ☐ Regular Undiscounted

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

- ☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application
- Registration Number 43861
- ☐ A sole inventor
- ☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application
- ☐ A joint inventor; all of whom are signing this request

Signature	/Robert Plotkin/
Name	Robert Plotkin

\*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).  
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

Electronic Patent Application Fee Transmittal				
<b>Application Number:</b>		13849606		
<b>Filing Date:</b>		25-Mar-2013		
<b>Title of Invention:</b>		Processing with Compact Arithmetic Processing Element		
<b>First Named Inventor/Applicant Name:</b>		Joseph Bates		
<b>Filer:</b>		Robert Plotkin		
<b>Attorney Docket Number:</b>		A0006-1001C2		
Filed as Small Entity				
<b>Filing Fees for Utility under 35 USC 111(a)</b>				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
<b>Basic Filing:</b>				
Statutory or Terminal Disclaimer	1814	1	160	160
<b>Pages:</b>				
<b>Claims:</b>				
<b>Miscellaneous-Filing:</b>				
<b>Petition:</b>				
<b>Patent-Appeals-and-Interference:</b>				
<b>Post-Allowance-and-Post-Issuance:</b>				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				160

Doc Code: DISQ.E.FILE

Document Description: Electronic Terminal Disclaimer – Approved

Application No.: 13849606

Filing Date: 25-Mar-2013

Applicant/Patent under Reexamination: Bates et al.

Electronic Terminal Disclaimer filed on July 24, 2015

☒ APPROVED

**This patent is subject to a terminal disclaimer**

☐ DISAPPROVED

Approved/Disapproved by: Electronic Terminal Disclaimer automatically approved by EFS-Web

U.S. Patent and Trademark Office



**Electronic Acknowledgement Receipt**

<b>EFS ID:</b>	23016926
<b>Application Number:</b>	13849606
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	6059
<b>Title of Invention:</b>	Processing with Compact Arithmetic Processing Element
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates
<b>Customer Number:</b>	24208
<b>Filer:</b>	Robert Plotkin
<b>Filer Authorized By:</b>	
<b>Attorney Docket Number:</b>	A0006-1001C2
<b>Receipt Date:</b>	24-JUL-2015
<b>Filing Date:</b>	25-MAR-2013
<b>Time Stamp:</b>	14:26:33
<b>Application Type:</b>	Utility under 35 USC 111(a)

**Payment information:**

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	\$ 160
RAM confirmation Number	730
Deposit Account	501797
Authorized User	PLOTKIN, ROBERT

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

## File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Electronic Terminal Disclaimer-Filed	eTerminal-Disclaimer.pdf	33343	no	2
			c8132334217a5c7e8f709965861a7446e5c34a63		

## Warnings:

## Information:

2	Fee Worksheet (SB06)	fee-info.pdf	30351	no	2
			d85bb9329e5c40869e317817f2b6a77c34419ecf		

## Warnings:

## Information:

**Total Files Size (in bytes):** 63694

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

### New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

### National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

### New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

ATTORNEY'S DOCKET NO: A0006-1001C2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial No.: 13/849,606  
Confirmation No.: 6059  
Title: Processing with Compact Arithmetic  
Processing Element  
First Inventor: BATES, Joseph  
Filing Date: Mar 25, 2013  
  
Examiner: Michael D. Yaary  
Art Unit: 2193  
  
Office Action Mailing Date: 01/26/2015

**RESPONSE TO NON-FINAL OFFICE ACTION**

In response to the Non-Final Office Action mailed on the Office Action Mailing Date indicated above, the following amendments and/or remarks are respectfully submitted:

**Remarks/Arguments** begin on page 2 of this paper.

**Conclusions** begin on page 3 of this paper.

**REMARKS/ARGUMENTS**

Claims 1, 16-18, and 33 stand rejected on the grounds of obviousness-type double-patenting over U.S. Pat. No. 8,407,273. Today Applicant filed a Terminal Disclaimer of this application over U.S. Pat. No. 8,407,273. Applicant, therefore, respectfully requests that the double patenting rejection be withdrawn.

All other pending claims have either already been deemed allowable by the Examiner or are dependent claims which depend on one of claims 1, 16-18, and 33. All claims, therefore, are now allowable.

**CONCLUSIONS**

Any dependent claims not specifically discussed above depend, either directly or indirectly, from the independent claims discussed above and therefore are patentable for at least the same reason(s).

If the Examiner wishes to discuss this Response, the Examiner is requested to call the Applicant's attorney at the phone number listed below.

No admission is made herein, explicitly or implicitly, that any amendments made herein are made for reasons of patentability.

If this response is not considered timely filed and if a request for extension of time is otherwise absent, applicant hereby requests any extension of time. Please charge any fees or make any credits, to Deposit Account No. 50/1797.

Respectfully submitted,

/Robert Plotkin/

Robert Plotkin, Esq.

Reg. No. 43,861

July 24, 2015

Date

Robert Plotkin, P.C.

15 New England Executive Park

Burlington, MA 01803 USA

Tel: (978) 318-9914

Fax: (978) 318-9060

## Electronic Patent Application Fee Transmittal

<b>Application Number:</b>	13849606			
<b>Filing Date:</b>	25-Mar-2013			
<b>Title of Invention:</b>	Processing with Compact Arithmetic Processing Element			
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates			
<b>Filer:</b>	Robert Plotkin			
<b>Attorney Docket Number:</b>	A0006-1001C2			
Filed as Small Entity				
<b>Filing Fees for Utility under 35 USC 111(a)</b>				
<b>Description</b>	<b>Fee Code</b>	<b>Quantity</b>	<b>Amount</b>	<b>Sub-Total in USD(\$)</b>
<b>Basic Filing:</b>				
<b>Pages:</b>				
<b>Claims:</b>				
<b>Miscellaneous-Filing:</b>				
<b>Petition:</b>				
<b>Patent-Appeals-and-Interference:</b>				
<b>Post-Allowance-and-Post-Issuance:</b>				
<b>Extension-of-Time:</b>				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension - 3 months with \$0 paid	2253	1	700	700
<b>Miscellaneous:</b>				
<b>Total in USD (\$)</b>				<b>700</b>

**Electronic Acknowledgement Receipt**

<b>EFS ID:</b>	23017172
<b>Application Number:</b>	13849606
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	6059
<b>Title of Invention:</b>	Processing with Compact Arithmetic Processing Element
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates
<b>Customer Number:</b>	24208
<b>Filer:</b>	Robert Plotkin
<b>Filer Authorized By:</b>	
<b>Attorney Docket Number:</b>	A0006-1001C2
<b>Receipt Date:</b>	24-JUL-2015
<b>Filing Date:</b>	25-MAR-2013
<b>Time Stamp:</b>	14:34:03
<b>Application Type:</b>	Utility under 35 USC 111(a)

**Payment information:**

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	\$ 700
RAM confirmation Number	892
Deposit Account	501797
Authorized User	PLOTKIN, ROBERT

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)



Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

## File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Amendment/Req. Reconsideration-After Non-Final Reject	Response.pdf	58585 a31c397854feb1b3faf2d324ada46a504ae8812b	no	3

## Warnings:

## Information:

2	Fee Worksheet (SB06)	fee-info.pdf	30383 63d6810b219bdfc37750a1abb4ebf7e8b3de9185	no	2
---	----------------------	--------------	---	----	---

## Warnings:

## Information:

**Total Files Size (in bytes):** 88968

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

### New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

### National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

### New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application Number	13/849,606
	Filing Date	3/25/2013
	First Named Inventor	Bates
	Art Unit	2193
	Client Number	A0006
Page 1 of 2	Matter Number	1001C2

FOREIGN PATENT DOCUMENTS					
Initials	Cite No.	Document Number	Publication Date	Country	T
	1	2008158822	07/10/2008	JP	
	2	2012530966	12/06/2012	JP	
	3	H0314128	01/22/1991	JP	

OTHER DOCUMENTS			
Initials	Cite No.	Author, Title, Date, Pages, etc.	T
	4	Naohito Nakasato, et al., "A Compiler for High Performance Adaptive Precision Computing," A paper of SACSIS (Symposium on Advanced Computing Systems and Infrastructures) 2008, Information Processing Society of Japan, June 4, 2008, No.5, pp. 149-156.	

Examiner Signature		Date Considered	
Examiner: Please initial if citation considered, whether or not citation is in conformance with MPEP Section 609. Please draw a line through the citation if it is not in conformance and it is not considered. Please include a copy of this form with the next communication to the applicant.			

<b>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</b>	<b>Application Number</b>	13/849,606
	<b>Filing Date</b>	3/25/2013
	<b>First Named Inventor</b>	Bates
	<b>Art Unit</b>	2193
	<b>Client Number</b>	A0006
Page 2 of 2	<b>Matter Number</b>	1001C2

<b>GENERAL</b>
<p>Pursuant to 37 C.F.R. 1.97 and 1.98 and to the duty of disclosure set forth in 37 C.F.R. 1.56, the Examiner in charge of the above-identified application is requested to consider and make of record the references listed herewith. A copy of each listed reference, other than U.S. patents/applications and references cited in a parent application, is enclosed.</p> <p>Although the information submitted herewith may be "material" to the Examiner's consideration of the subject application, this submission is not intended to constitute an admission that such information is "prior art" as to the claimed invention.</p> <p>In accordance with 37 C.F.R. 1.97(g), the filing of this Information Disclosure Statement shall not be construed to mean that a search has been made.</p>

<b>TIMING</b>
<p>In accordance with 37 CFR 1.97(c), this Information Disclosure Statement is being filed after the period specified in 37 CFR 1.97(b) and before the mailing date of a Final Action under 37 CFR 1.113, a Notice of Allowance under 37 CFR 1.311, or an Action that otherwise closes prosecution in the application.</p>

<b>CERTIFICATION STATEMENT</b>
<p>Each item of information contained in this information disclosure statement was first cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this information disclosure statement.</p>

<b>FEE</b>
<p>No fee is required. This Information Disclosure Statement is being filed in accordance with 37 CFR 1.97(c), and is accompanied by the certification statement of 37 CFR 1.97(e)(1).</p> <p>If necessary, the Director is hereby authorized to charge or credit Deposit Account No. 50-1797 for any additional fees, or any underpayment or credit for overpayment in connection herewith. Please reference attorney docket number A0006-1001C2 for any such charge or credit.</p>

<b>SIGNATURE</b>			
Signature	/Robert Plotkin/	Date	June 25, 2015
Name	Robert Plotkin	Registration Number	43861

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2008-158822

(P2008-158822A)

(43) 公開日 平成20年7月10日(2008.7.10)

(51) Int. Cl.	F I	テーマコード (参考)
G06F 7/38 (2006.01)	G06F 7/38 V	5B016
G06F 9/30 (2006.01)	G06F 9/30 380Z	5B033
	G06F 9/30 380R	

審査請求 未請求 請求項の数 5 O L (全 18 頁)

(21) 出願番号 特願2006-346997 (P2006-346997)

(22) 出願日 平成18年12月25日 (2006.12.25)

(出願人による申告) 平成17年度、文部科学省、「科学技術試験研究」委託研究、産業再生法第30条の適用を受ける特許出願

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番1号

(74) 代理人 100074099

弁理士 大西 義之

(74) 代理人 100067987

弁理士 久木元 彰

(72) 発明者 安藤 壽茂

神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

Fターム(参考) 5B016 AA02 BA03 BA04 BB06 CA01

CB02 CD01 HA01

5B033 FA08 FA14

(54) 【発明の名称】 演算回路及び演算方法並びに情報処理装置

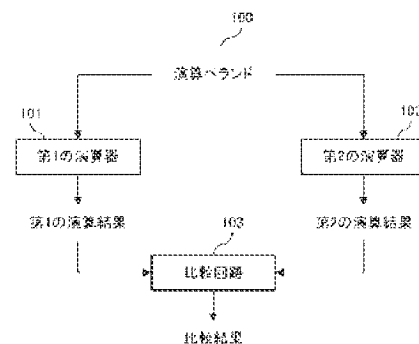
(57) 【要約】

【課題】より少ない回路量で数値的に大きな誤差をもつエラーを効率的に検出する浮動小数点演算回路を提供する。

【解決手段】上記課題を解決するために、浮動小数点演算回路に、第1の演算結果を出力する第1の演算器101と、第2の演算結果を出力する第2の演算器102と、第1の演算結果と第2の演算結果とについて所定ビット幅の比較を行う比較回路103と、を備える。

【選択図】 図1

本発明の実施例に係る  
浮動小数点演算回路の概要を説明する図



## 【特許請求の範囲】

## 【請求項 1】

浮動小数点形式で表現された演算オペランドを入力し、該浮動小数点形式における仮数部に第 1 のデータ幅を有する第 1 の演算結果を出力する第 1 の演算器と、

前記演算オペランドを入力し、前記浮動小数点形式における仮数部に前記第 1 のデータ幅よりも小さい第 2 のデータ幅を有する第 2 の演算結果を出力する第 2 の演算器と、

前記第 1 の演算結果と前記第 2 の演算結果の仮数部について、それぞれの所定ビットから前記第 2 のデータ幅について比較を行う比較回路とを有することを特徴とする浮動小数点演算回路。

## 【請求項 2】

10

前記比較回路は、

前記第 1 の演算結果と前記第 2 の演算結果の仮数部について、最上位ビットから前記第 2 のデータ幅についてビット毎に排他的論理和演算を行なう排他的論理和演算回路と、

該排他的論理和演算の演算結果について、所定のビットパターンであるか否かを判定し、該所定のビットパターンの場合に一致の比較結果を出力する仮数部比較回路と、を有し、

該仮数部比較回路が一致の比較結果を出力した場合に、前記第 1 の演算結果と前記第 2 の演算結果とが一致すると判断することを特徴とする請求項 1 記載の浮動小数点演算回路。

## 【請求項 3】

20

前記所定のビットパターンは、

全てのビットが 1 のビットパターン、全てのビットが 0 のビットパターン、または、最上位ビットから任意のビットまで 0 が連続し該任意のビットの次のビットから所定のビットまで 1 が連続するビットパターンのいずれかであることを特徴とする請求項 2 記載の浮動小数点演算回路。

## 【請求項 4】

浮動小数点形式で表現された演算オペランドを入力し、該浮動小数点形式における仮数部に第 1 のデータ幅を有する第 1 の演算結果を出力する第 1 の演算器に接続され、前記第 1 の演算結果の検査を行う演算検査回路において、

前記演算オペランドを入力し、前記浮動小数点形式における仮数部に前記第 1 のデータ幅よりも小さい第 2 のデータ幅を有する第 2 の演算結果を出力する第 2 の演算器と、

前記第 1 の演算結果と前記第 2 の演算結果の仮数部について、それぞれの所定ビットから前記第 2 のデータ幅について比較を行う比較回路とを有することを特徴とする演算検査回路。

## 【請求項 5】

30

浮動小数点形式で表現された演算オペランドを入力し、該浮動小数点形式における仮数部に第 1 のデータ幅を有する第 1 の演算結果を出力する第 1 の演算回路と、

前記演算オペランドを入力し、前記浮動小数点形式における仮数部に前記第 1 のデータ幅よりも小さい第 2 のデータ幅を有する第 2 の演算結果を出力する第 2 の演算回路と、

前記第 1 の演算結果と前記第 2 の演算結果の仮数部について、それぞれの所定ビットから前記第 2 のデータ幅について比較を行う比較回路とを有することを特徴とする情報処理装置。

40

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、浮動小数点演算時に発生するエラーを訂正することができる演算回路及び演算方法並びに情報処理装置に関する。

## 【背景技術】

## 【0002】

半導体製造プロセスの進歩にともなう微細化により、例えば、半導体に用いる配線幅の

50



縮小化も進んでいる。その結果、半導体の高集積化が進む一方で、回路の誤作動の確率も増加している。

【0003】

特に、大規模な科学技術計算を行うスーパーコンピュータでは、多数の浮動小数点演算器が用いられ、アルファ線や宇宙線起因の中性子の衝突により浮動小数点演算器の一つで誤動作が発生し計算結果が誤る危険性が増加する。

【0004】

例えば、浮動小数点演算器1個の故障率を10FIT (Failure In Time: 1億時間に1回の故障が発生する率)としても、100万個の浮動小数点演算器を用いるスーパーコンピュータでは、100時間に1回の頻度で、いずれかの浮動小数点演算器にエラーが発生することになる。

10

【0005】

浮動小数点演算器のエラーを検出する方法としては、2つの同一の演算器を並列に動作させ、両者の結果を比較する方法がある。しかし、通常の2倍の演算器や比較回路が必要となるので回路量が大幅に増加してしまい、多数の浮動小数点演算器を必要とするスーパーコンピュータにとっては負担が大きくなってしまう。

【0006】

また、加算器ではパリティ予測、乗算器ではModulo 3剰余チェックを行うなどの方法により、浮動小数点演算器の主要部分のエラーを検出することもできる。しかし、パリティチェックでは偶数個の出力ビットが誤るエラーは検出できない。また、Modulo 3剰余チェックでは同一の剰余となるエラーは検出できない。さらに、これらのチェックを行うためには演算器自体の20%を超える量の回路をエラー検出のために追加しなくてはならない。

20

【0007】

特許文献1には、丸め・桁合わせ器の出力と下位mビットの出力を選択的に出力して観測することにより、2進乗算器の下位に位置するゲートの故障を検出しやすくする浮動小数点演算器について開示されている。

【特許文献1】特開平06-083591号公報

【発明の開示】

【発明が解決しようとする課題】

30

【0008】

以上に説明したように、なるべく少ない回路量で、浮動小数点演算で問題となるエラーを高い確率で検出を行なう演算回路及び計算機システムが望まれており、さらに、エラーが検出された場合には、その演算命令を再度実行することにより中性子の衝突などに起因する間欠エラーを訂正できる演算回路及び計算機システムが望まれている。

【0009】

本発明は、上述した問題に鑑みてなされたものであり、その解決しようとする課題は、より少ない回路量で数値的に大きな誤差をもつエラーを効率的に検出することである。

【課題を解決するための手段】

【0010】

40

上記課題を解決するために、本発明に係る浮動小数点演算回路は、浮動小数点形式で表現された演算オペランドを入力し、該浮動小数点形式における仮数部に第1のデータ幅を有する第1の演算結果を出力する第1の演算器と、前記演算オペランドを入力し、前記浮動小数点形式における仮数部に前記第1のデータ幅よりも小さい第2のデータ幅を有する第2の演算結果を出力する第2の演算器と、前記第1の演算結果と前記第2の演算結果の仮数部について、それぞれの所定ビットから前記第2のデータ幅について比較を行う比較回路と、を有する。

【0011】

本発明によると、比較回路が第1の演算結果と第2の演算結果の仮数部について、それぞれの所定ビットから第2のデータ幅について比較を行うので、比較の結果に応じて第1

50

の演算結果が正しいか否かを判定することが可能となる。

【0012】

また、第2の演算器は、記浮動小数点形式における仮数部に第1のデータ幅よりも小さい第2のデータ幅を有する第2の演算結果を出力する演算器なので、より少ない回路量で第1の演算結果が正しいか否かを判定することが可能となる。

【0013】

さらに、第2の演算結果は、第1のデータ幅よりも小さい第2のデータ幅を有するので、第1の演算結果と第2の演算結果とで一定以上に大きい誤差がある場合には比較の結果から確実に検出することが可能となる。

【発明の効果】

10

【0014】

以上に説明したように、本発明によると、より少ない回路量で数値的に大きな誤差をもつエラーを効率的に検出することが可能となる。

【発明を実施するための最良の形態】

【0015】

以下、本発明の実施の形態について図1～図10に基づいて説明する。

図1は、本発明の実施例に係る浮動小数点演算回路100の概要を説明する図である。

図1に示す浮動小数点演算回路100は、第1の演算結果を出力する第1の演算器101と、第2の演算結果を出力する第2の演算器102と、第1の演算結果と第2の演算結果とについて所定ビット幅だけ比較する比較回路103と、を備える。

20

【0016】

第1の演算器101は、浮動小数点形式で表現された演算オペランドが入力されると、第1の演算結果を出力する。第1の演算結果は浮動小数点形式のデータであり、その仮数部は第1のデータ幅（例えば、52ビット幅）を有する。以下、この第1の演算結果を構成する符号部、指数部及び仮数部を、それぞれ第1の符号部、第1の指数部及び第1の仮数部という。ただし、本実施例に係る第1の符号部、指数部及び仮数部は、丸め処理が施されていないものとする。

【0017】

第2の演算器102は、第1の演算器101と同じ演算オペランドが入力される。そして、第2の演算結果を出力する。第2の演算結果も浮動小数点形式のデータであり、その仮数部には第2のデータ幅を有する。ここで、第2のデータ幅は、第1のデータ幅よりも小さいデータ幅（例えば、4ビット幅）とする。したがって、第2の演算器102は、第1の演算器101より回路量が少ない。

30

【0018】

以下、この第2の演算結果を構成する符号部、指数部及び仮数部を、それぞれ第2の符号部、第2の指数部及び第2の仮数部という。

比較回路103は、第1の演算結果と第2の演算結果とについて第2のデータ幅だけ比較し、比較結果を出力する。

【0019】

以下、本発明の実施例に係る浮動小数点演算回路100の具体例として倍精度浮動小数点演算回路について説明する。ただし、本発明を倍精度浮動小数点演算回路に限定する趣旨ではない。例えば、単精度浮動小数点演算回路等についても本発明は適用可能なのは当然である。

40

【0020】

なお、以下の説明では、本実施例に係る浮動小数点演算回路100への入力データは、IEEE (Institute of Electrical and Electronic Engineers) 754に基づく倍精度浮動小数点形式のデータとする。そして、符号部S1 (1ビット幅)と指数部E1 (11ビット幅)と仮数部F1 (52ビット幅)とで構成される第1の演算オペランドと、符号部S2 (1ビット幅)と指数部E2 (11ビット幅)と仮数部F2 (52ビット幅)とで構成される第2の演算オペランドを

50

入力データとする。

【0021】

したがって、本実施例に係る浮動小数点演算回路100（すなわち、第1の演算器101）の出力も、IEEE754に基づく倍精度浮動小数点形式のデータであり、符号部SIGN（1ビット）と指数部EXP（11ビット幅）と仮数部FRAC（52ビット幅）とで構成される出力データとする。

【0022】

（第1の実施例）

図2は、本発明の第1の実施例に係る第1の演算器101の具体的な構成例を示す図である。

10

【0023】

図2に示す第1の演算器101は、仮数部F1とF2の和を求める加算器201と、指数部E1とE2の差を求める引算器202と、指数部E1とE2の差を絶対値に変換する変換器203と、その絶対値に応じて仮数部をシフトする右シフタ204と、指数部E1とE2の差の正負に応じて出力を切替える切替器205～207と、符号部S1とS2の排他的論理和を求める排他的論理和演算器208と、加算器201の出力値を絶対値に変換する変換器209と、最上位ビットからみて最初に1を持つビット位置を検出するビット位置検出器210と、検出したビット位置が最上位となるように左シフトを行うビット位置調整器211と、丸め処理を行う丸め処理器212と、符号部S1とS2と加算器201および引算器202の出力値の符号とから符号部SIGNを決定するデコーダ213と、指数部E1またはE2とビット位置検出器210およびビット位置調整器211との差を求める引算器214と、を備える倍精度浮動小数点加算器である。

20

【0024】

加算処理では、指数部E1とE2の桁を合わせる必要がある。そのため、引算器202は、指数部E1とE2の差を求める。そして、その結果に応じて、右シフタ203は仮数部F2を右シフトする。

【0025】

ここで、指数部E1とE2の差が負の値となる場合には、仮数部F1を右にシフトする必要がある。引算器202の結果の正負に応じて、切替器205及び206の出力を切替えて、右シフタ203及び加算器201への入力（仮数部F1またはF2）を切替える。

30

【0026】

加算器201は、2つの演算オペランドの仮数部（52ビット）に最上位の“1”を復元して53ビット幅のデータの加算を行い、54ビット幅のデータを出力する加算器である。なお、加算器201は、排他的論理和演算器201の出力値に応じて加算または引算を行う。符号部S1とS2が同じ場合は加算を行い、異なる場合は引算を行う。

【0027】

仮数部F1とF2は常に正の値であるので、変換器209は、加算器201の出力値の絶対値をとって正の数に揃える。ビット位置検出器210は、正の数となった加算結果の最上位ビットからみて最初に“1”を持つビット位置を検出し、ビット位置調整器211に出力する。

40

【0028】

ビット位置調整器211は、変換器209から入力される加算結果を左シフトし、ビット位置検出器210が検出したビット位置を最上位ビットの位置にシフトする。この左シフト分を指数部EXPに反映するために、ビット位置検出器210が検出したビット位置は、引算器214にも入力される。

【0029】

丸め処理器212は、ビット位置調整器211から出力された値について丸め処理を行う。なお、この丸め処理は、IEEE754に基づく丸め処理であるので詳細は省略する。

50



## 【0030】

引算器214は、仮数部(E1またはE2)からビット位置検出器210の出力値及び丸め処理器212によるシフト分を差し引いて桁合わせを行う。

デコーダ213は、符号部(S1、S2)と引算器202の出力値の符号と加算器201の出力値の符号とから符号部SIGNを計算する。

## 【0031】

以上の処理によって、第1の演算オペランドと第2の演算オペランドとの加算結果(符号部SIGN、指数部EXP、仮数部Frac)が求められる。また、本実施例では、デコーダ213、切替器207および変換器209の出力値を、それぞれ第1の符号部、第1の指数部および第1の仮数部として使用する。

10

## 【0032】

図3は、本発明の第1の実施例に係る第2の演算器102及び比較回路103の具体的な構成例を示す図である。

なお、第2の演算器102には、第1の演算器101と同様に、第1及び第2の演算オペランドが入力されるが、仮数部は上位4ビットのみを使用する。図3の説明では、符号部S1(1ビット幅)と指数部E1(11ビット幅)と仮数部f1(4ビット幅)とで構成される第1の演算オペランドと、符号部S2(1ビット幅)と指数部E2(11ビット幅)と仮数部f2(4ビット幅)とで構成される第2の演算オペランドを入力データとする。

## 【0033】

20

図3に示す第2の演算器102は、仮数部f1とf2の和を求める加算器301と、指数部E1とE2の差を求める引算器202と、指数部E1とE2の差を絶対値に変換する変換器203と、その絶対値に応じて仮数部をシフトする右シフタ302と、指数部E1とE2の差の正負に応じて出力を切替える切替器205～207と、符号部S1とS2の排他的論理和を求める排他的論理和演算器208と、加算器201の出力値を絶対値に変換する変換器303と、を備える倍精度浮動小数点加算器である。

## 【0034】

第1の演算器101と同様に、右シフタ302は、引算器202が算出する指数部E1とE2の差に応じて仮数部f2を右シフトする。また、引算器202の出力値の正負に応じて、切替器205及び206の出力を切替えて、加算器301及び右シフタ302への入力(仮数f1またはf2)を切替える。

30

## 【0035】

加算器301は、例えば、5ビット幅のデータを出力する加算器である。なお、加算器301は、第1の演算器と同様に、排他的論理和演算器208の信号に応じて、加算または引算が行われる。変換器303は、加算器301の出力値の絶対値をとって正の数に換える。

## 【0036】

デコーダ213は、符号部S1とS2と引算器202及び加算器201の出力値の符号とから第2の符号部を算出する。

以上の処理によって、第1の演算オペランドと第2の演算オペランドとの加算処理が行われ、第2の演算結果が算出される。本実施例では、デコーダ213、切替器207および変換器303の出力値を、それぞれ第2の符号部、第2の指数部および第2の仮数部として使用する。

40

## 【0037】

図3に示す比較回路103は、第1の仮数部と第2の仮数部とを比較して誤差が一定範囲内か否かを判定する略一致検出器304と、第1の指数部と第2の指数部とを比較して不一致を検出する不一致検出器305と、第1の符号部と第2の符号部とを比較して不一致を検出する不一致検出器306と、略一致検出器304、不一致検出器305及び306の出力値について論理和を求めてエラーを検出する論理和演算器307と、を備える。

## 【0038】

50

第1の仮数部と第2の仮数部は、略一致検出器304に入力される。ここで、第2の演算結果における仮数部は、その最下位1ビット(1LSB: Least Significant Bit)分のエラーを含むことは避けられない。そのため、正しい精度で計算が行われた場合でも1LSBだけ異なる場合がある。以下、この時の誤差を「1LSBの誤差」という。

#### 【0039】

そこで、本実施例に係る比較回路103では、完全な一致検出回路ではなく、1LSBの誤差を許容する比較を行う略一致検出器304を使用する。なお、誤差の範囲を大きくする場合、例えば、最下位2ビット分のエラー分を許容する場合には、略一致検出器304の入力データを1ビット減じてLSBを除く上位ビットを比較することにより2LSBの誤差を許容する比較を行うこともできる。詳しくは図5で後述する。

10

#### 【0040】

第1の仮数部と第2の仮数部は、略一致検出器304に入力される。略一致検出器304は、両仮数部の誤差が一定範囲内にあるか否かを判定する。例えば、誤差が一定範囲(1LSBの誤差の範囲)内にある場合には一致と判定し、一定範囲(1LSBの誤差の範囲)内にはない場合には不一致と判定する。なお、具体的な構成例については、図6で説明する。

#### 【0041】

第1の指数部と第2の指数部は、不一致検出器305に入力される。不一致検出器305は、ビットごとの排他的論理和をとり、その出力にたいして論理和演算を行うことによって両指数部の一致・不一致を判定する。同様に、第1の符号部と第2の符号部は、不一致検出器306に入力される。そして、不一致検出器306は、排他的論理和演算を行うことによって両符号部の一致・不一致を判定する。

20

#### 【0042】

略一致検出器304、不一致検出器305及び306の出力は、論理和演算器307に入力される。例えば、略一致検出器304、不一致検出器305及び306がそれぞれ不一致を検出した場合に1を出力すると、論理和演算器307は、略一致検出器304、不一致検出器305及び306の少なくとも1つ以上が不一致を検出した場合に1を出力する。その結果、第1の演算結果と第2の演算結果との不一致、すなわち、第1の演算器101のエラーを検出することができる。

30

#### 【0043】

図4は、本発明の第1の実施例に係る略一致検出器304が行う1LSBの誤差を許容する比較を説明する図である。

図4(a)は、略一致検出器304に入力されるAとBとがLSB以外全く同じ場合を示している。例えば、Aが“11110”、Bが“11111”のような場合である。A及びBの排他的論理和をとると、その演算結果Cは、LSB以外が全て“0”(0の連続)となり、LSBのみ“1”となる。

#### 【0044】

図4(b)は、略一致検出器304に入力されるAとBとが1LSB異なる場合を示している。例えば、Aが“10000”、Bが“01111”のように数値的には1LSBのみ異なる場合である。A及びBの排他的論理和をとると、その演算結果Cは、AとBとが一致したビット位置では“0”(0の連続)となり、一致しないビット位置では“1”(1の連続)となる。

40

#### 【0045】

図4(a)及び(b)のいずれの場合も、排他的論理和をとった演算結果Cは、上位ビットから順に“0”が連続し、“1”が現われるとその後は“1”が連続するビットパターンとなっていることが分かる。すなわち、AとBとが上述のビットパターンとなっていれば、AとBとは1LSB以下の違いであると判定できる。

#### 【0046】

なお、演算結果Cが全て“1”のビットパターンの場合には、AとBとの誤差が1LSB

50

分の誤差の場合であり、演算結果Cが全て"0"のビットパターンの場合は、AとBとが完全一致の場合であるので、AとBとが1LSB以下の違いである場合に含まれる。

【0047】

以下、図5～図6を用いて比較回路103の具体的な構成例について説明する。

図5は、本発明の第1の実施例に係る比較回路103の入力部の構成例を示す図である。A0～Anは、ビット幅がnである場合の第1の仮数部の各ビットを示し、B0～Bnは、ビット幅がnである場合の第2の仮数部の各ビットを示している。なお、本実施例の場合には、n=4とすればよい。

【0048】

比較回路103の入力部には、排他的論理和演算回路500～50nを備え、A0～AnとB0～Bnについて、それぞれ排他的論理和をとって演算結果X0～Xnを出力する。図5に示すように、比較回路103を1LSBの誤差を許容する比較回路とする場合には、A0とB0、A1とB1、・・・、AnとBnの排他的論理和の演算結果を、それぞれX0、X1、・・・Xnとし、比較回路103を2LSBの誤差を許容する比較回路とする場合には、A1とB1、A2とB2、・・・、AnとBnの排他的論理和の演算結果を、それぞれX0、X1、・・・Xn-1とし、"0"をXnとする。

【0049】

図6は、本発明の第1の実施例に係る比較回路103の判定部の構成例を示す図である。なお、説明を簡単にするために8ビット精度(n=7)の比較回路について説明するが、これに限定する趣旨でないのは当然である。

【0050】

図5に示した演算結果X0とX1、X2とX3、X4とX5、X6とX7は、それぞれ論理積演算器601～604と、論理和演算器611～614に入力される。

論理積演算器601と602の出力O10とO32は、論理積演算器605に入力される。論理積演算器605は、出力O30を出力する。同様に、論理積演算器603と604の出力O54とO76は、論理積演算器606に入力される。論理積演算器606は、出力O74を出力する。

【0051】

したがって、出力Onmはmビット目からnビット目までが"1"であることを示す。

また、論理和演算器611と612の出力Z10とZ32は、論理積演算器615に入力される。論理積演算器615は、出力Z30を出力する。同様に、論理和演算器613と614の出力Z54とZ76は、論理積演算器616に入力される。論理積演算器616は、出力Z74を出力する。

したがって、出力Znmはmビット目からnビット目までが"0"であることを示す。そして、以上の処理から、X0～X7における"0"の連続と"1"の連続が求められる。

【0052】

さらに、出力Z30とZ74、Z32とZ74とX1、O10とZ74とX2、O30とZ76とX5の反転、O30とO54とX7の反転、O30とO74は、それぞれ論理積演算器621～626に入力される。論理積演算器621～626の出力は、反転器629～634を介して論理積演算器635に入力される。

【0053】

以上の構成において、例えば、(1)O30=O74=1の場合には、X0～X7が全て1であると判定できる。また、(2)X7=0、かつ、O54=O30=1、(3)Z76=X5=0、かつ、O30=1、(4)Z74=0、かつ、X2=O10=1、(5)Z74=Z32=0、かつ、X1=1の場合には、X0～X7が上位から所定ビットまで0が連続し続いて1が連続すると判定できる。さらに、(6)Z74=Z30=0の場合には、X0～X7が全て0である(AとBとが完全に一致する)と判定できる。

【0054】

以上の(1)～(6)の場合に、論理積演算器635は"1"を出力する。すなわち、

AとBとの誤差が1LSBより大きい場合に"1"を出力する。

その結果、比較回路103の出力(0または1)から、比較回路103に入力される第1及び第2の仮数部の誤差が1LSBの範囲である略一致か否かを判定することが可能となる。

【0055】

(第2の実施例)

図7は、本発明の第2の実施例に係る第1の演算器101の具体的な構成例を示す図である。

【0056】

図7に示す第1の演算器101は、仮数部F1とF2の積を求める乗算器701と、丸め処理を行う丸め処理器212と、指数部E1とE2の和から指数部を算出する加算器702と、符号部S1とS2の排他的論理和から符号部を算出する排他的論理和演算器703と、を備える倍精度浮動小数点乗算器である。

10

【0057】

乗算器701は、2つの演算オペランドの仮数部(52ビット)に最上位の"1"を復元し、53ビット幅のデータの掛け算を行う。また、加算器702は、2つの演算オペランドの指数部の足し算を行う。排他的論理和演算器703は、2つの演算オペランドの符号部の排他的論理和を求める。

【0058】

乗算器701による乗算結果は、丸め処理器212でIEEE規格にしたがった丸め処理が行われる。その結果、桁上がりが発生した場合には、加算器704で補正が行われる。

20

【0059】

以上の処理によって、第1の演算オペランドと第2の演算オペランドとの乗算結果(符号部SIGN、指数部EXP、仮数部FRAC)が求められる。また、本実施例では、排他的論理和演算器703、加算器702および乗算器701の出力値を、それぞれ第1の符号部、第1の指数部および第1の仮数部として使用する。

【0060】

図8は、本発明の第2の実施例に係る第2の演算器102及び比較回路103の具体的な構成例を示す図である。

30

なお、第1の実施例と同様に、第2の演算器102には、第1及び第2の演算オペランドが入力されるが、仮数部は上位4ビットのみを使用する。図8の説明では、符号部S1(1ビット幅)と指数部E1(11ビット幅)と仮数部f1(4ビット幅)とで構成される第1の演算オペランドと、符号部S2(1ビット幅)と指数部E2(11ビット幅)と仮数部f2(4ビット幅)とで構成される第2の演算オペランドを入力データとする。

【0061】

図8に示す第2の演算器102は、仮数部f1とf2の積を求める乗算器801と、指数部E1とE2の和から第2の指数部を算出する加算器702と、符号部S1とS2の排他的論理和から第1の符号部を算出する排他的論理和演算器804と、を備える倍精度浮動小数点乗算器である。

40

【0062】

乗算器801は、例えば、4+1(第2の仮数部のビット幅+1)ビットの掛け算を行う乗算器である。乗算器801は、2つの演算オペランドの仮数部(4ビット)に最上位の"1"を復元し、5ビット幅のデータの掛け算を行う。

【0063】

加算器702は、2つの演算オペランドの指数部と(-1023)の足し算を行う。排他的論理和演算器804は、2つの演算オペランドの符号部と第1の符号部との排他的論理和を求める。

【0064】

以上の処理によって、第1の演算オペランドと第2の演算オペランドとの乗算処理が行

50



われ、第2の演算結果が算出される。本実施例では、排他的論理和演算器804、加算器702および乗算器801の出力値を、それぞれ第2の符号部、第2の指数部および第2の仮数部として使用する。

【0065】

図8に示す比較回路103は、第1の仮数部と第2の仮数部とを比較して誤差が一定範囲（2LSBの誤差の範囲）内か否かを判定する略一致検出器802と、第1の指数部と第2の指数部とを比較して不一致を検出する不一致検出器803と、第1の符号部と第2の符号部とを比較して不一致を検出する不一致検出器804と、略一致検出器802、不一致検出器803及び804の出力結果について論理和を求めてエラーを検出する論理和演算器805と、を備える。なお、第2の演算器102で使用する排他的論理和演算器804は、比較回路103で使用する不一致検出器804と共有しているが、別々に構成してもよい。

【0066】

第1の仮数部と第2の仮数部は、略一致検出器802に入力される。略一致検出器802は、第1の仮数部と第2の仮数部との誤差が一定範囲（2LSBの誤差の範囲）内にあるか否かを判定する。

【0067】

ここで、例えば、第1の演算オペランドaと第2の演算オペランドbとにそれぞれ誤差（ $\Delta a$ 、 $\Delta b$ ）が含まれる場合の乗算は、下記の式を計算することになる。

$$a * (1 - \Delta a) * b * (1 - \Delta b) = a * b * (1 - \Delta a - \Delta b + \Delta a * \Delta b)$$

今、第1及び第2の仮数部が4ビットであるとする、 $\Delta a$ 、 $\Delta b$ は $1/32$ より小さい値となる。しかし、乗算結果の誤差（ $\Delta a + \Delta b - \Delta a * \Delta b$ ）は、 $1/16$ より小さい値となるが加算結果の誤差のように $1/32$ より小さい値とならない。そこで、本実施例に係る略一致検出器802では、2LSBの誤差を許容する比較を行う略一致検出器を使用する。なお、具体的な構成例は、図5及び図6で説明したので省略する。

【0068】

第1の指数部と第2の指数部は、不一致検出器803に入力される。不一致検出器803は、ビットごとに排他的論理和をとり、それらに対して論理和演算を行うことによって両指数部の一致・不一致を判定する。第1の符号部と第1及び第2の演算オペランドの符号部S1、S2は、排他的論理和演算器804に入力され、排他的論理和が求められる。

【0069】

論理和演算器805は、略一致検出器802、不一致検出器803及び804の論理和をとる。したがって、例えば、略一致検出器802、不一致検出器803及び804がそれぞれ不一致を検出した場合に1を出力すると、論理和演算器307は、略一致検出器304、不一致検出器305及び306の少なくとも1つ以上が不一致を検出した場合に1を出力する。その結果、第1の演算結果と第2の演算結果との不一致、すなわち、第1の演算器101のエラーを検出することができる。

【0070】

第一の演算器と第二の演算器で符号と指数の計算回路は同じ規模の回路を二組必要とするが、次に述べるように、第二の演算器の仮数部の回路量は大幅に少なくなる。

一般に、加算器やシフタの回路量は、ビット幅をNとすると、およそ

【0071】

【数1】

$$N \cdot \log_2 N$$

【0072】

に比例する。したがって、例えば、52ビットの加算器201や右シフタ204と比べて、5ビットの加算器301や右シフタ302に必要な回路量は約 $1/20$ である。

したがって、第1の実施例においては、仮数部が52ビットの場合は、浮動小数点演算

10

20

30

40

50

回路100全体の10%以下の回路量で第2の演算器102及び比較回路103を実現することができる。

【0073】

また、乗算器の回路量はおおよそ $N$ の2乗に比例する。したがって、例えば、53ビットの乗算器701に比べて、5ビットの乗算器801は $1/100$ 程度の回路量で実現することができる。

【0074】

したがって、第2の実施例においては、浮動小数点演算回路100全体の2～3%の回路量で第2の演算器102及び比較回路103を実現することができる。

なお、仮数部は省略された1.0を補うと1.0以上2.0未満である。そこで、仮数部の値を1.0とみなして第1及び第2の符号部と第1及び第2の指数部についてのみ比較を行うようにすれば、エラーを検出する精度は低下するが、例えば、加算器301や乗算器801が不要となるため必要となるハードウェアを更に減少させることができる。

【0075】

図9は、本実施例を反復集束計算に適応した場合の例を説明する図である。

図9に示すグラフは、2次方程式： $Y = X * X - 3 * X + 2$ のグラフである。以下、反復収束計算であるNewton-Raphson法を用いてこの2次元方程式の解を求める場合について説明する。

【0076】

2次方程式 $Y(X) = X * X - 3 * X + 2$ は、 $X = 1$ と $X = 2$ で $Y = 0$ となるが、これを $X = 0$ からスタートして $X = 1$ の解を求めることを考える。

$dY/dX = 2 * X - 3$ であるので、 $X = 0$ における $Y$ の傾き $Y'(0) = -3$ 、 $Y(0) = 2$ となる。この時、 $X = X - Y(0)/Y'(0)$ から次の $X$ の値を求めると $X = 2/3$ となる。同様に、 $X = 2/3$ における傾き $Y'(2/3) = -5/3$ 、 $Y(2/3) = 4/9$ となる。次の $X$ の値は $X = 14/15$ となるので、同様に、 $Y'(14/15) = -17/15$ 、 $Y(14/15) = 16/225$ となる。その次の $X$ の値は $X = 254/255$ となる。この時、正しい解 $X = 1$ との誤差は、0.4%程度に減少する。

【0077】

以上の計算から明らかなように、 $Y$ や $Y'$ または $X$ の補正量の計算に計算エラーが生じて次の $X$ に誤差が生じたとしても、 $X$ が1.5未満に留まる限り反復を繰返すと $X = 1.0$ に収束することになる。

【0078】

なお、計算エラーで誤差が生じて解の値から遠ざかってしまうと反復回数が増加し計算時間が余計にかかることになるが、上述した中性子などによるエラーの発生頻度は低いので全体の計算時間に与える影響は無視できる程度である。

【0079】

一方、計算エラーによる誤差が大きく、 $X$ が1.5より大きくなってしまえば以降の反復で $X$ の値は2.0に収束することになる。これも $X * X - 3 * X + 2 = 0$ の解であるので、この2次方程式の解としては正しいが、実際の反復収束計算では、中間での計算誤差が大きいと、意図しない点に収束してしまったり、高次のカーブでは収束点が見つからず発散してしまう場合がある。

【0080】

例えば、スーパーコンピュータ等の主要な用途となるナノやバイオ分野で用いられる第一原理計算では、各電子間の反発力を計算し、それを合計して各原子に働く力を計算し、それにより原子の位置を動かし、また、各原子に働く力を計算する、という反復により、系全体のエネルギーが最小に収束するように計算を行う。

【0081】

各原子に働く力は、他の原子全ての電子からの和であるので、一つの電子から受ける力の計算において、ハードウェアの間欠エラーが原因である程度の誤差が生じたとしても、数千、数万の電子の一つであるので、その影響は軽減される。

## 【0082】

このように、一般的な計算では、個々の計算において桁が大きく異なるようなエラーが発生しなければ、値として小さなエラーの影響は反復収束過程で自動的に修正されてしまうケースが多い。

## 【0083】

したがって、本実施例に係る浮動小数点演算回路100を用いて桁違いに大きな誤差となるエラーを検出して再実行を行うことによって、アルファ線や宇宙線起因の中性子の衝突等の固定障害でない障害に起因して発生する間欠エラーを訂正することが可能となる。

## 【0084】

図10は、本発明の実施例に係る浮動小数点演算回路100の再実行に必要な構成の例を示す図である。なお、ハードウェアエラーが検出された場合の再実行に関しては、以下の文献に詳細が記載されているので、概要について説明する。

## 【0085】

1. H. Ando, T. Kitamura, M. Shebanow, M. Butler, US Patent 6, 519, 730, "Computer and error recovery method for the same"

2. H. Ando, Y. Yoshida, A. Inoue, I. Sugiyama, T. Asakawa, K. Morita, T. Muta, T. Motokurumada, S. Okada, H. Yamashita, Y. Satsukawa, A. Konmoto, R. Yamashita, H. Sugiyama, "A 1.3-GHz fifth-generation SPARC64 microprocessor", Solid-State Circuits, IEEE Journal, Volume 38, 11. 11. 2003, 1896-1905

図10に示す浮動小数点演算回路1000は、演算回路の実行に必要な状態を保持するプロセッサ・ステイト1001と、正しく演算が実行された時の状態を格納するチェックポイント・アレイ1002と、最後に正しく実行を完結した状態を格納しているエントリを指すコミット・ポインタ1003と、直前の状態を格納するエントリを指すカレント・ポインタ1004と、誤作動等が起こった場合にコミット・ポインタで指された状態をチェックポイント・アレイ1002から取り出し、プロセッサ・ステイト1001に格納することにより、その状態から実行を再開させる制御を行うバックアップ・コントロール1005と、分岐命令を取り扱うブランチ・ユニット1006と、各ユニットの誤作動を監視するエラー検出ユニット1007と、各ユニットを制御するコントロール・ユニット1008と、第1の演算器101で構成された演算実行ユニット1009と、第2の演算器102及び比較回路103で構成することにより演算実行ユニット1009の演算結果の正誤を監視するエラーチェックユニット1010と、を少なくとも備える。

## 【0086】

分岐先が決定する前に、方向を予測して投機的に実行を行うには、その分岐命令の実行直前の状態（プロセッサ・ステイト1001の内容）をカレント・ポインタ1004が指すチェックポイント・アレイ1002のエントリに格納する。そして、予測した分岐先の命令実行を開始する。そして、ブランチ・ユニット1006は、カレント・ポインタ1004を一つ進める。また、命令の実行完結を監視し、コミット・ポインタ1003の指すエントリの次のエントリまで実行が完結すると、コミット・ポインタ1003を一つ進め、常に最後の実行命令の完結状態を指すように制御する。

## 【0087】

分岐予測ミスが発見された場合は、条件分岐命令の直前のチェックポイントをチェックポイント・アレイ1002から取り出してプロセッサ・ステイトを復元してチェックポイント直後の命令から実行を再開することにより、分岐予測の誤りを修正する。

## 【0088】

第1及び第2の実施例で示した処理によって、エラー検出ユニット1007がエラーを検出すると、エラー検出ユニット1007は、バックアップ・コントロール1005に誤

動作を通知する。

【0089】

誤作動の通知を受けると、バックアップ・コントロール1005は、コミット・ポイント1003で指される正しく実行を完結した状態を取り出し、プロセッサ・ステイト1001に復元する。そして、この状態から実行が再開される（再実行される）。その結果、中性子の衝突などに起因する間欠エラーを訂正することができる。この手順は、分岐予測誤りの修正と同様の手順であり、分岐予測ミスのために存在する回路を有効に利用することが出来る。

【0090】

以上に説明したように、第1の演算器101に短い精度（例えば、4ビット）の第2の演算器102と比較回路103とをエラーチェック用に追加し、第1の演算結果と第2の演算結果とが一定の誤差（例えば、1LSBの誤差）を許容する比較を行うことにより大きな数値誤差となるエラーを効率的に検出することが可能となる。さらに、検出したエラーは上述した再実行の機構を用いて訂正を行うことが可能となる。

【0091】

また、本実施例に係る第2の仮数部は4ビット幅であるので、第2の演算器102及び比較回路103は、浮動小数点演算器の場合にあっては、浮動小数点演算回路100全体の2～3%程度の回路量、浮動小数点加算器の場合でも浮動小数点演算回路100全体の10%以下の回路量で実現することが可能となる。すなわち、従来の手法に比べてエラー検出に必要な回路量を低減することが可能となる。

【0092】

また、演算結果の数値が何桁も異なるような、パリティチェックや剰余チェックにより浮動小数点演算器の主要部分のエラーを検出する従来の方法では検出できない間欠エラーであっても、本発明の方法では、第2の仮数部の精度より大きな誤差を確実に検出することが可能となる。

【0093】

また、反復集束計算を行う場合には、許容される数値誤差以下のエラーはその反復収束計算の過程で訂正されるので、従来のエラー検出法に比べて少ない追加回路でエラー訂正回路を実現でき、かつ、計算が収束しない、または、異常を検出せずに終了して計算結果にエラーが生じ等の問題が発生する確率を大幅に低減することが可能となる。

【0094】

（付記1） 浮動小数点形式で表現された演算オペランドを入力し、該浮動小数点形式における仮数部に第1のデータ幅を有する第1の演算結果を出力する第1の演算器と、

前記演算オペランドを入力し、前記浮動小数点形式における仮数部に前記第1のデータ幅よりも小さい第2のデータ幅を有する第2の演算結果を出力する第2の演算器と、

前記第1の演算結果と前記第2の演算結果の仮数部について、それぞれの所定ビットから前記第2のデータ幅について比較を行う比較回路とを有することを特徴とする浮動小数点演算回路。

（付記2） 前記比較回路は、

前記第1の演算結果と前記第2の演算結果の仮数部について、最上位ビットから前記第2のデータ幅についてビット毎に排他的論理和演算を行なう排他的論理和演算回路と、

該排他的論理和演算の演算結果について、所定のビットパターンであるか否かを判定し、該所定のビットパターンの場合に一致の比較結果を出力する仮数部比較回路と、を有し、

該仮数部比較回路が一致の比較結果を出力した場合に、前記第1の演算結果と前記第2の演算結果とが一致すると判断することの特徴とする付記1記載の浮動小数点演算回路。

（付記3） 前記所定のビットパターンは、

全てのビットが1のビットパターン、全てのビットが0のビットパターン、または、最上位ビットから任意のビットまで0が連続し該任意のビットの次のビットから所定のビットまで1が連続するビットパターンのいずれかであることを特徴とする付記2記載の浮動



小数点演算回路。

(付記4) 前記第1の演算結果と前記第2の演算結果について、それぞれの符号を比較し、該比較の結果が一致する場合に一致の比較結果を出力する符号比較回路をさらに有し、

前記仮数部比較回路と前記符号比較回路とが一致の比較結果を出力した場合に、前記第1の演算結果と前記第2の演算結果とが一致すると判断することを特徴とする付記2記載の浮動小数点演算回路。

(付記5) 前記第1の演算結果と前記第2の演算結果について、それぞれの指数部を比較し、該比較の結果が一致する場合に一致の比較結果を出力する指数部比較回路をさらに有し、

10

前記仮数部比較回路と前記指数部比較回路とが一致の比較結果を出力した場合に、前記第1の演算結果と前記第2の演算結果とが一致すると判断することを特徴とする付記2記載の浮動小数点演算回路。

(付記6) 前記比較回路の比較の結果に応じて前記第1の演算回路に対して演算を再実行させる再実行回路をさらに有することを特徴とする付記1記載の浮動小数点演算回路。

(付記7) 浮動小数点形式で表現された演算オペランドを入力し、該浮動小数点形式における仮数部に第1のデータ幅を有する第1の演算結果を出力する第1の演算器に接続され、前記第1の演算結果の検査を行う演算検査回路において、

前記演算オペランドを入力し、前記浮動小数点形式における仮数部に前記第1のデータ幅よりも小さい第2のデータ幅を有する第2の演算結果を出力する第2の演算器と、

20

前記第1の演算結果と前記第2の演算結果の仮数部について、それぞれの所定ビットから前記第2のデータ幅について比較を行う比較回路とを有することを特徴とする演算検査回路。

(付記8) 前記比較回路は、

前記第1の演算結果と前記第2の演算結果の仮数部について、最上位ビットから前記第2のデータ幅についてビット毎に排他的論理和演算を行なう排他的論理和演算回路と、

該排他的論理和演算の演算結果について、所定のビットパターンであるか否かを判定し、該所定のビットパターンの場合に一致の比較結果を出力する仮数部比較回路と、を有し、

該仮数部比較回路が一致の比較結果を出力した場合に、前記第1の演算結果と前記第2の演算結果とが一致すると判断することを特徴とする付記7記載の演算検査回路。

30

(付記9) 前記所定のビットパターンは、

全てのビットが1のビットパターン、全てのビットが0のビットパターン、または、最上位ビットから任意のビットまで0が連続し該任意のビットの次のビットから所定のビットまで1が連続するビットパターンのいずれかであることを特徴とする付記8記載の演算検査回路。

(付記10) 前記第1の演算結果と前記第2の演算結果について、それぞれの符号を比較し、該比較の結果が一致する場合に一致の比較結果を出力する符号比較回路をさらに有し、

前記仮数部比較回路と前記符号比較回路とが一致の比較結果を出力した場合に、前記第1の演算結果と前記第2の演算結果とが一致すると判断することを特徴とする付記8記載の演算検査回路。

40

(付記11) 前記第1の演算結果と前記第2の演算結果について、それぞれの指数部を比較し、該比較の結果が一致する場合に一致の比較結果を出力する指数部比較回路をさらに有し、

前記仮数部比較回路と前記指数部比較回路とが一致の比較結果を出力した場合に、前記第1の演算結果と前記第2の演算結果とが一致すると判断することを特徴とする付記8記載の演算検査回路。

(付記12) 前記比較回路の比較の結果に応じて前記第1の演算回路に対して演算を再実行させる再実行回路をさらに有することを特徴とする付記7記載の演算検査回路。

50

(付記 13) 浮動小数点形式で表現された演算オペランドを入力し、該浮動小数点形式における仮数部に第 1 のデータ幅を有する第 1 の演算結果を出力する第 1 の演算回路と、前記演算オペランドを入力し、前記浮動小数点形式における仮数部に前記第 1 のデータ幅よりも小さい第 2 のデータ幅を有する第 2 の演算結果を出力する第 2 の演算回路と、前記第 1 の演算結果と前記第 2 の演算結果の仮数部について、それぞれの所定ビットから前記第 2 のデータ幅について比較を行う比較回路とを有することを特徴とする情報処理装置。

(付記 14) 前記比較回路は、

前記第 1 の演算結果と前記第 2 の演算結果の仮数部について、最上位ビットから前記第 2 のデータ幅についてビット毎に排他的論理和演算を行なう排他的論理和演算回路と、該排他的論理和演算の演算結果について、所定のビットパターンであるか否かを判定し、該所定のビットパターンの場合に一致の比較結果を出力する仮数部比較回路と、を有し、

10

該仮数部比較回路が一致の比較結果を出力した場合に、前記第 1 の演算結果と前記第 2 の演算結果とが一致すると判断することを特徴とする付記 13 記載の情報処理装置。

(付記 15) 前記所定のビットパターンは、

全てのビットが 1 のビットパターン、全てのビットが 0 のビットパターン、または、最上位ビットから任意のビットまで 0 が連続し該任意のビットの次のビットから所定のビットまで 1 が連続するビットパターンのいずれかであることを特徴とする付記 14 記載の情報処理装置。

20

(付記 16) 前記第 1 の演算結果と前記第 2 の演算結果について、それぞれの符号を比較し、該比較の結果が一致する場合に一致の比較結果を出力する符号比較回路をさらに有し、

前記仮数部比較回路と前記符号比較回路とが一致の比較結果を出力した場合に、前記第 1 の演算結果と前記第 2 の演算結果とが一致すると判断することを特徴とする付記 14 記載の情報処理装置。

(付記 17) 前記第 1 の演算結果と前記第 2 の演算結果について、それぞれの指数部を比較し、該比較の結果が一致する場合に一致の比較結果を出力する指数部比較回路をさらに有し、

前記仮数部比較回路と前記指数部比較回路とが一致の比較結果を出力した場合に、前記第 1 の演算結果と前記第 2 の演算結果とが一致すると判断することを特徴とする付記 14 記載の情報処理装置。

30

(付記 18) 前記比較回路の比較の結果に応じて前記第 1 の演算回路による演算を再実行する再実行回路をさらに有することを特徴とする付記 13 記載の情報処理装置。

(付記 19) 第 1 の演算器が出力する第 1 の演算結果を、第 2 の演算器が出力する第 2 の演算結果を用いて検査を行う演算回路の演算方法であって、

浮動小数点形式で表現された演算オペランドを入力し、該浮動小数点形式における仮数部に第 1 のデータ幅を有する第 1 の演算結果を出力するステップと、

前記演算オペランドを入力し、前記浮動小数点形式における仮数部に前記第 1 のデータ幅よりも小さい第 2 のデータ幅を有する第 2 の演算結果を出力するステップと、

40

前記第 1 の演算結果と前記第 2 の演算結果の仮数部について、それぞれの所定ビットから前記第 2 のデータ幅について比較を行うステップとを有することを特徴とする浮動小数点演算方法。

(付記 20) 浮動小数点形式で表現された演算オペランドを入力し、浮動小数点形式における仮数部に第 1 のデータ幅を有する第 1 の演算結果を出力する第 1 の演算器に接続され、前記第 1 の演算結果の検査を行う演算検査回路の検査方法であって、

前記演算オペランドを入力し、前記浮動小数点形式における仮数部に前記第 1 のデータ幅よりも小さい第 2 のデータ幅を有する第 2 の演算結果を出力するステップと、

前記第 1 の演算結果と前記第 2 の演算結果の仮数部について、それぞれの所定ビットから前記第 2 のデータ幅について比較を行うステップとを有することを特徴とする演算検査

50

方法。

【図面の簡単な説明】

【0095】

【図1】本発明の実施例に係る浮動小数点演算回路の概要を説明する図である。

【図2】本発明の第1の実施例に係る第1の演算器の具体的な構成例を示す図である。

【図3】本発明の第1の実施例に係る第2の演算器及び比較回路の具体的な構成例を示す図である。

【図4】本発明の第1の実施例に係る略一致検出器が行う1LSBの誤差を許容する比較を説明する図である。

【図5】本発明の第1の実施例に係る比較回路の入力部の構成例を示す図である。

10

【図6】本発明の第1の実施例に係る比較回路の入力部の構成例を示す図である。

【図7】本発明の第2の実施例に係る第1の演算器の具体的な構成例を示す図である。

【図8】本発明の第2の実施例に係る第2の演算器及び比較回路の具体的な構成例を示す図である。

【図9】本実施例を反復集束計算に適応した場合の例を説明する図である。

【図10】本発明の実施例に係る浮動小数点演算回路の再実行に必要な構成の例を示す図である。

【符号の説明】

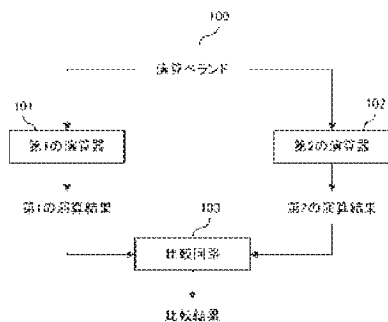
【0096】

100	浮動小数点演算回路
101	第1の演算器
102	第2の演算器
103	比較回路

20

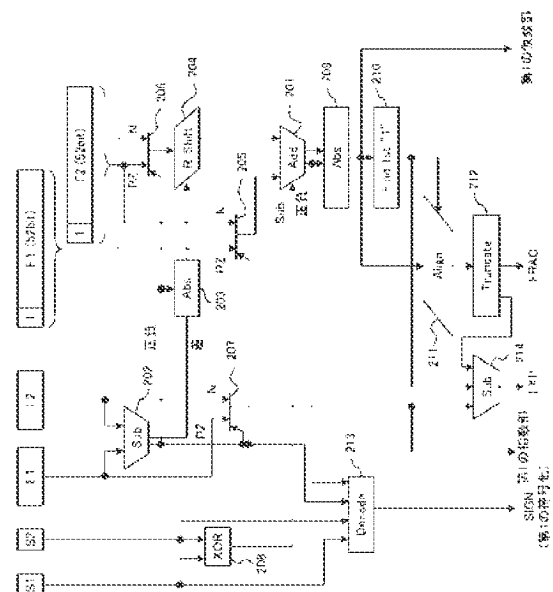
【図1】

本発明の実施例に係る  
浮動小数点演算回路の概要を説明する図



【図2】

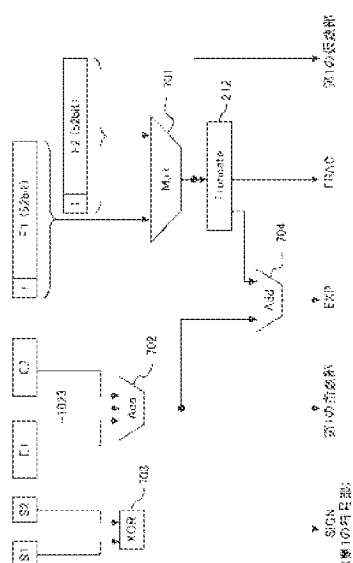
本発明の第1の実施例に係る  
第1の演算器の具体的な構成例を示す図



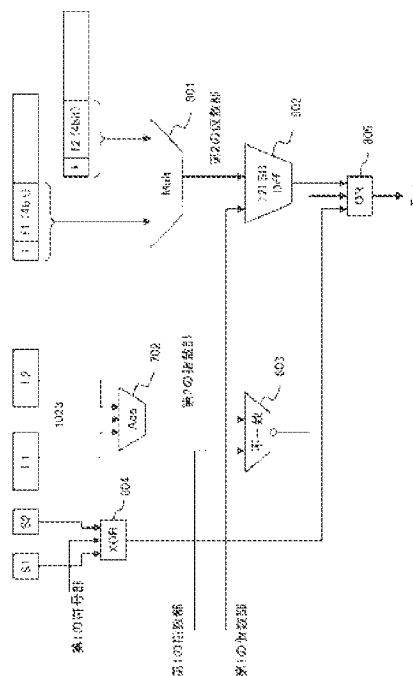


[X] [ ]

本発明の第2の実施例に係る  
第1の演算器の具体的な構成例を示す図

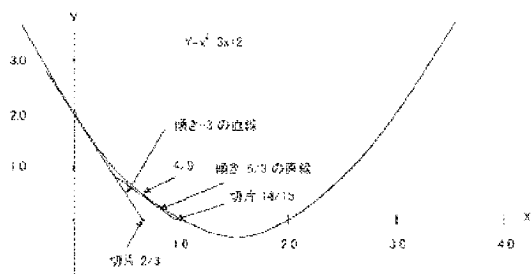


本発明の第2の実施例に係る第2の演算器及び比較回路の具体的な構成例を示す図



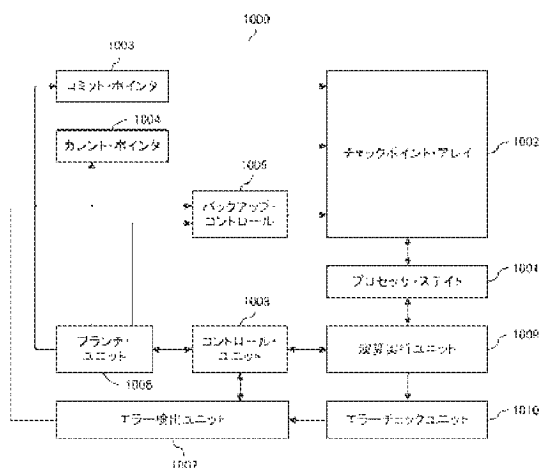
【 9 】

本実施例を反復集束計算に  
適応した場合の例を説明する図



【☒ 10】

本発明の実施例に係る浮動小数点演算回路の再実行に必要な構成の例を示す図





US 2008/0155004A1

(19) **United States**(12) **Patent Application Publication**  
**ANDO**(10) **Pub. No.: US 2008/0155004 A1**(43) **Pub. Date: Jun. 26, 2008**(54) **ARITHMETIC CIRCUIT, ARITHMETIC  
METHOD, AND INFORMATION  
PROCESSING DEVICE**(30) **Foreign Application Priority Data**

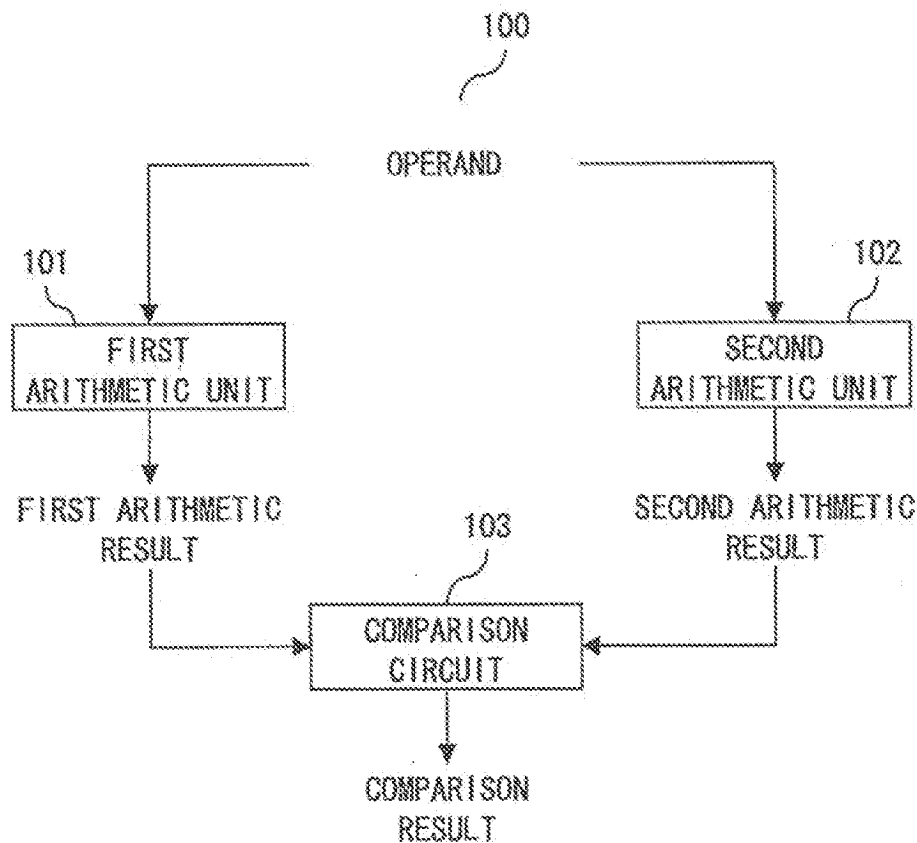
Dec. 25, 2006 (JP) ..... 2006-346997

**Publication Classification**(51) **Int. Cl.**  
**G06F 7/02** (2006.01)(52) **U.S. Cl.** ..... 708/495(57) **ABSTRACT**

To provide a floating point arithmetic circuit for efficiently detecting an error, which has a large numerical error, with a less circuit amount, the floating point arithmetic circuit comprises a first arithmetic unit for outputting a first arithmetic result, a second arithmetic unit for outputting a second arithmetic result, and a comparison circuit for making a comparison between the first and the second arithmetic results by a predetermined bit width.

(75) **Inventor:** Hisashige ANDO, Kawasaki (JP)

Correspondence Address:  
STAAS & HALSEY LLP  
SUITE 700, 1201 NEW YORK AVENUE, N.W.  
WASHINGTON, DC 20005

(73) **Assignor:** FUJITSU LIMITED, Kawasaki (JP)(21) **Appl. No.:** 11/864,084(22) **Filed:** Sep. 28, 2007



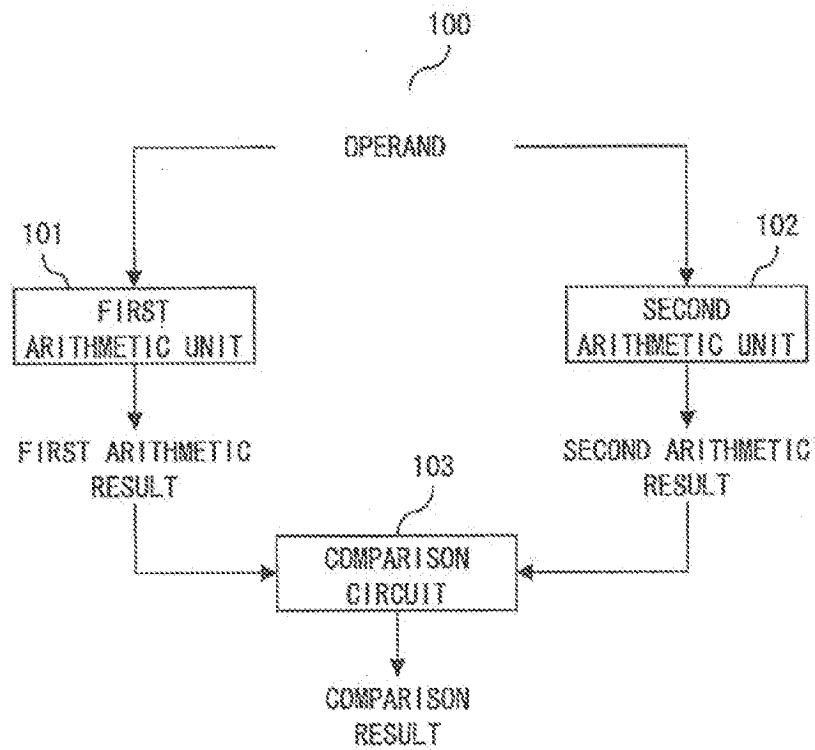
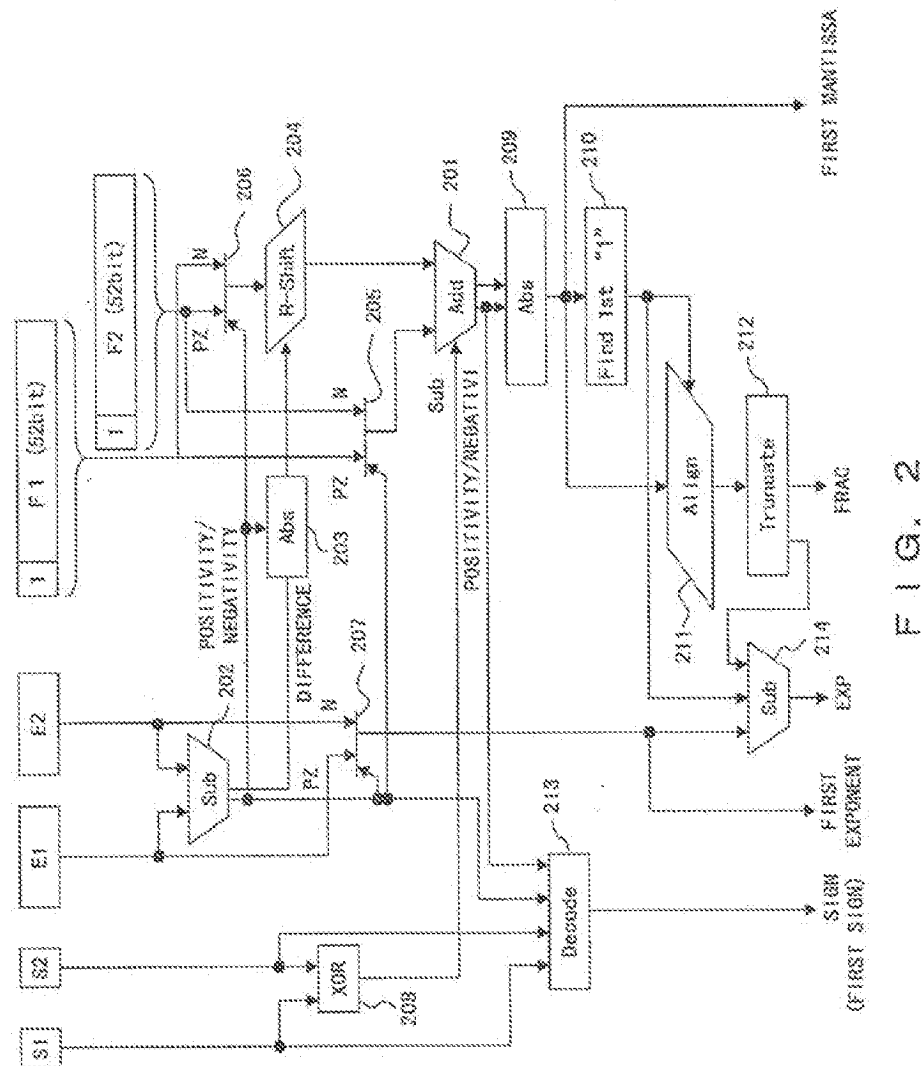


FIG. 1





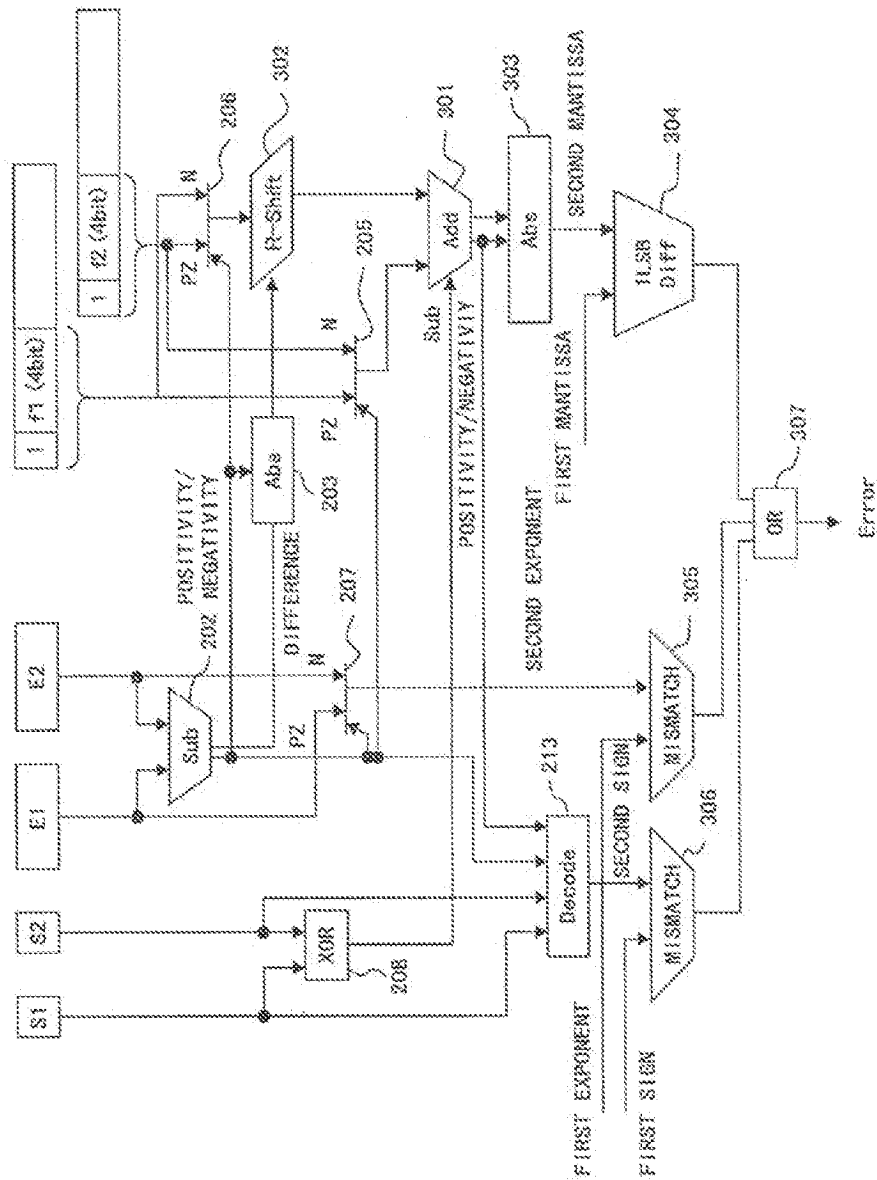


FIG. 3

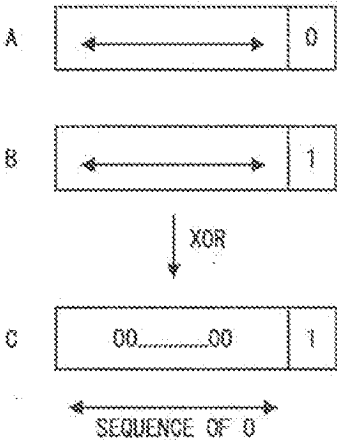


FIG. 4A

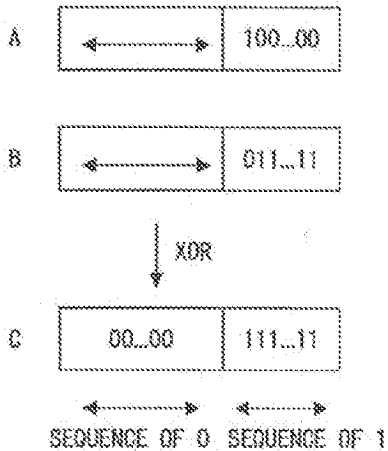


FIG. 4B

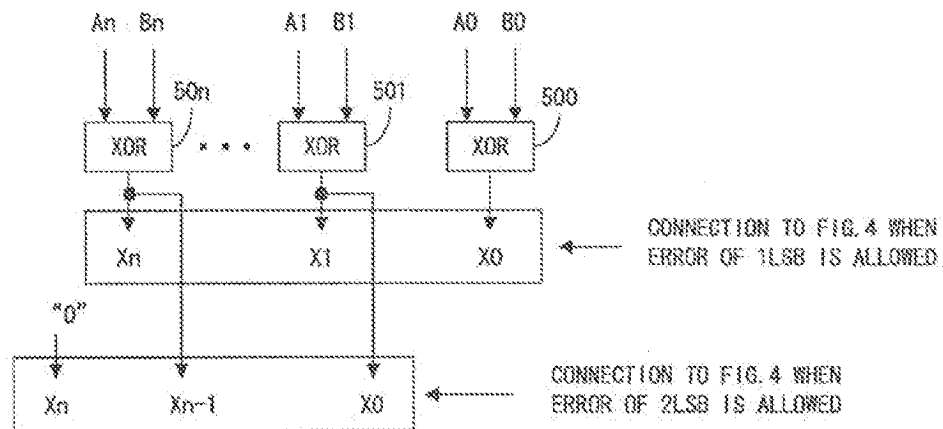


FIG. 5

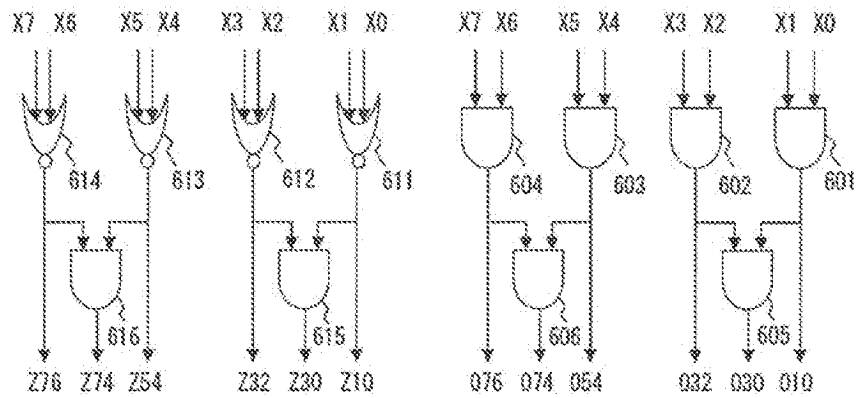


FIG. 6A

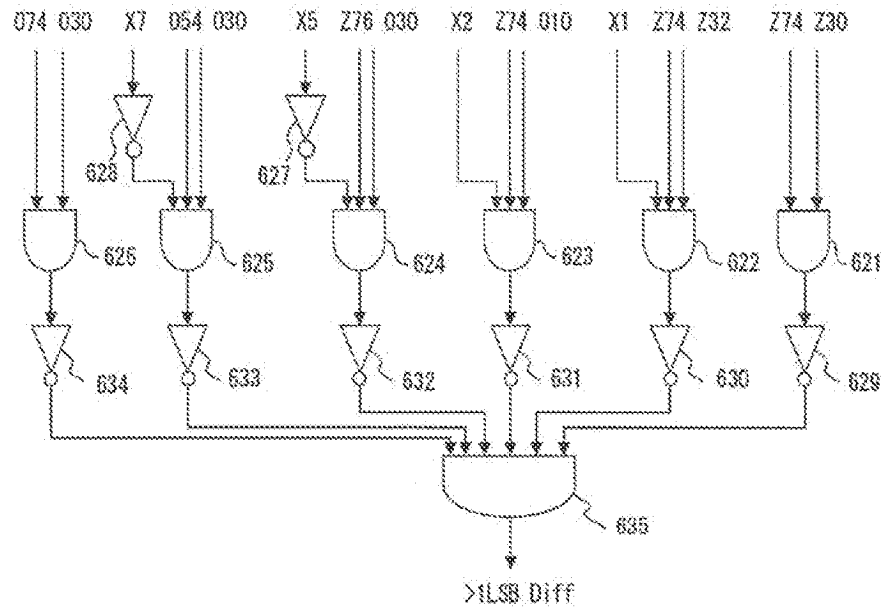


FIG. 6B

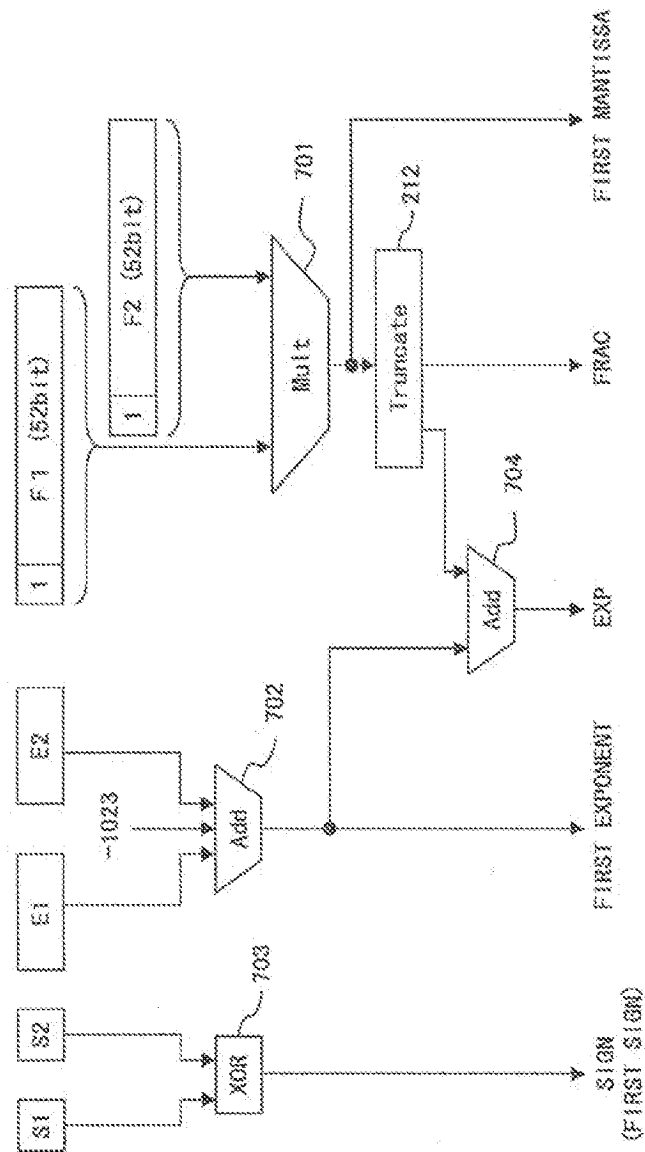


FIG. 7

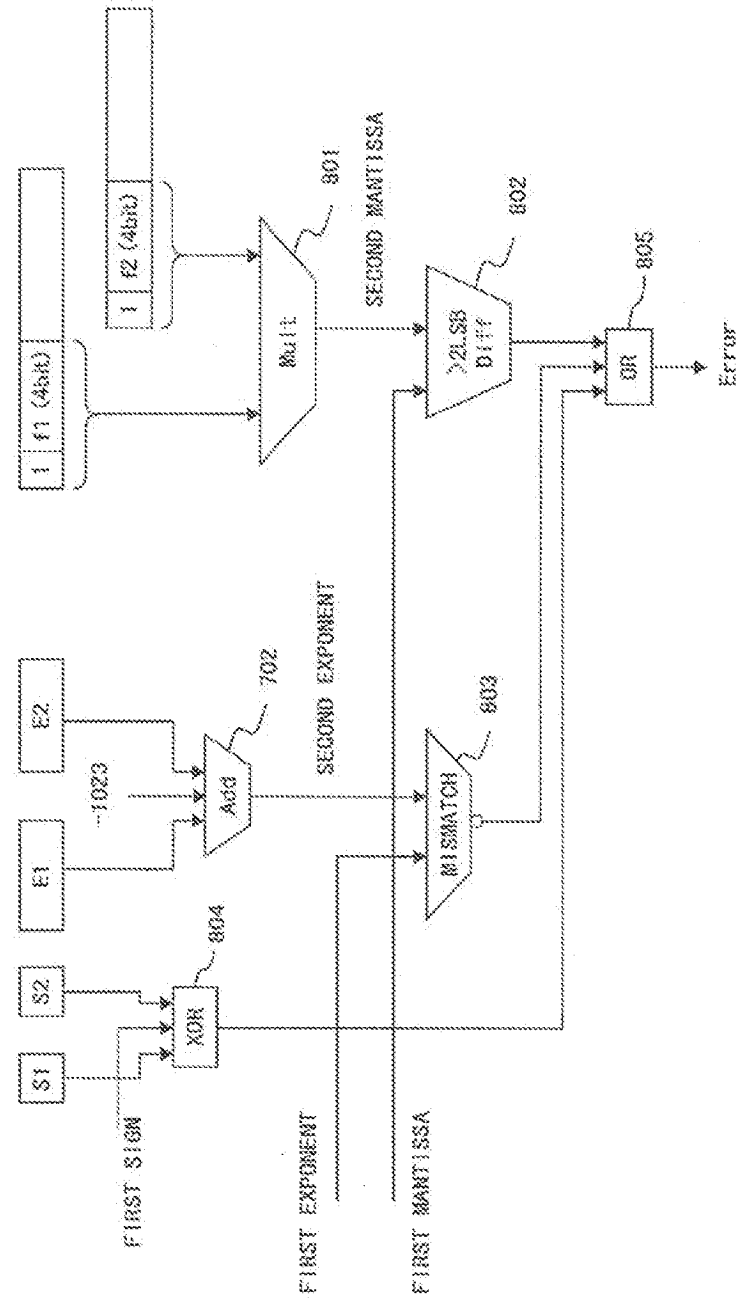


FIG. 8

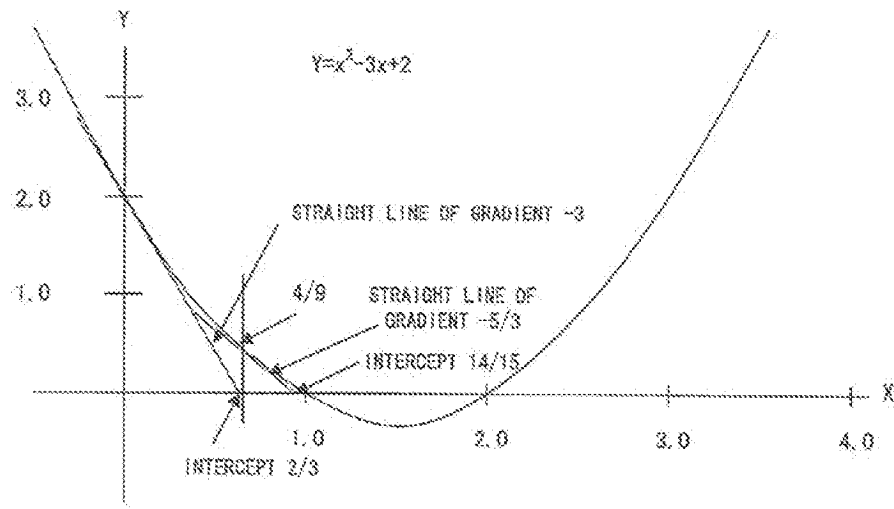


FIG. 9

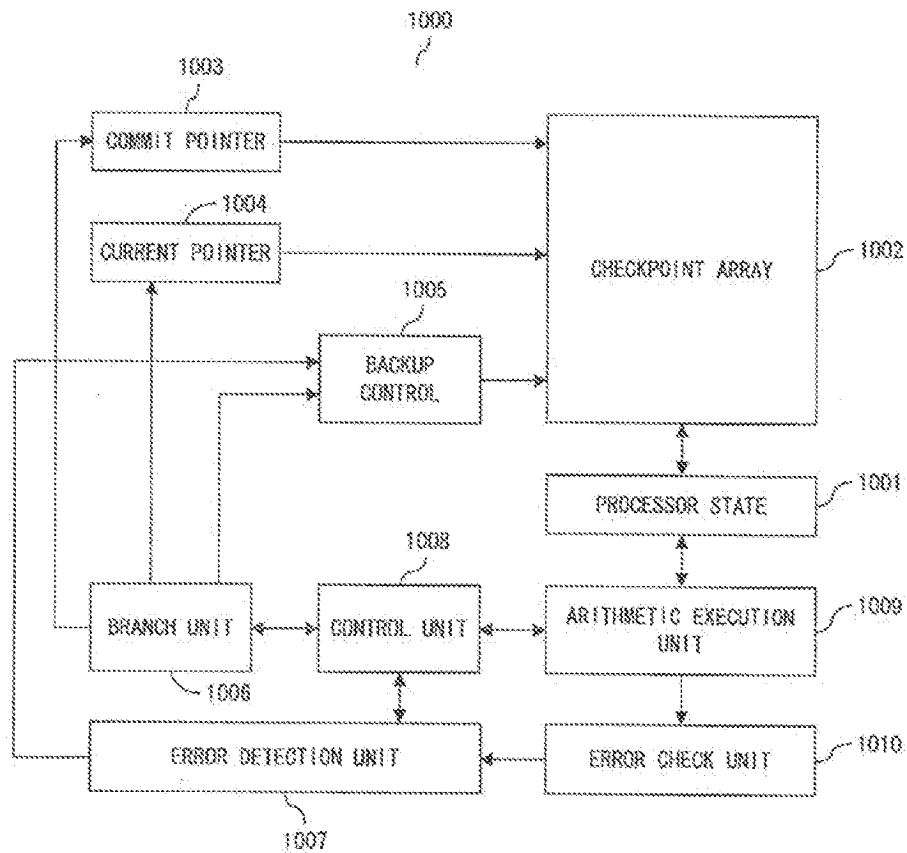


FIG. 10



US 2008/0155004 A1

Jun. 26, 2008

1

# ARITHMETIC CIRCUIT, ARITHMETIC METHOD, AND INFORMATION PROCESSING DEVICE

## BACKGROUND OF THE INVENTION

[0001] 1. Field of the Invention

[0002] The present invention relates to an arithmetic circuit, an arithmetic method, and an information processing device, which can correct an error that occurs at the time of a floating point arithmetic.

[0003] 2. Description of the Related Art

[0004] For finer-line processes with the advance of a semiconductor manufacturing process, by way of example, also reductions in a wire width used for a semiconductor are in progress. As a result, the higher-level integration of a semiconductor proceeds, but at the same time, the probability of misoperation of a circuit increases.

[0005] Especially, in a supercomputer that performs a large-scale scientific technical calculation, many floating point arithmetic units are used, and a risk such that a misoperation occurs in one of the floating point arithmetic units due to the collision of radiation particles, which is caused by an alpha ray or a cosmic ray, and a calculation result becomes incorrect increases.

[0006] For example, if the failure rate of one floating point arithmetic unit is assumed to be 10 FIT (Failure In Time: rate at which one failure occurs per one hundred million hours), an error occurs in any of the floating point arithmetic units with a frequency of once per one hundred hours in a super computer that uses one million floating point arithmetic units.

[0007] As a method for detecting an error of a floating point arithmetic unit, a method for causing two identical arithmetic units to run in parallel, and for making a comparison between results of both of the arithmetic units exists. However, since this method requires arithmetic units and comparison circuits the numbers of which are twice as many as usual, a circuit amount significantly increases, and a burden on a supercomputer, which requires many floating point arithmetic units, becomes heavy.

[0008] Additionally, also an error of a principal portion of a floating point arithmetic unit can be detected also with a method such as parity prediction in an adder, and Modulo 3 residue check in a multiplier, and the like. However, with the parity check, an error cannot be detected when the number of error bit is even. Furthermore, with the Modulo 3 residue check, an error of the same residue cannot be detected. Still further, a circuit amount of 20 percent or more of an arithmetic unit itself must be added to detect an error in order to make these checks.

[0009] Japanese Published Patent Application No. H6-083591 discloses a floating point arithmetic unit that facilitates the detection of a failure of a gate positioned in a low order of a binary multiplier by selectively making and observing the output of a round off/digit aligner, and the output of low-order in bits.

[0010] As described above, an arithmetic circuit and a computer system, which detect an error that is problematic in a floating point arithmetic, with high probability and with a circuit amount as least as possible is desired, and besides, an arithmetic circuit and a computer system, which can correct an intermittent error caused by the collision of radiation par-

ticles, etc. by reexecuting an arithmetic instruction upon detection of an error, are desired.

## SUMMARY OF THE INVENTION

[0011] The present invention was developed in light of the above described problems, and an object thereof is to efficiently detect an error, which has a large numerical error, with a less circuit amount.

[0012] To achieve the above described object, a floating point arithmetic circuit according to the present invention comprises a first arithmetic unit, to which an operand represented in a floating point format is input, for outputting a first arithmetic result having a first data width in a mantissa of the floating point format, a second arithmetic unit, to which the operand is input, for outputting a second arithmetic result having a second data width, which is smaller than the first data width, in a mantissa of the floating point format, and a comparison circuit for making a comparison between the mantissas of the first and the second arithmetic results by the second data width from each predetermined bit.

[0013] According to the present invention, the comparison circuit makes a comparison between the mantissas of the first and the second arithmetic results by the second data width from each predetermined bit, whereby whether or not the first arithmetic result is correct can be determined according to the result of the comparison.

[0014] Additionally, the second arithmetic unit is an arithmetic unit for outputting the second arithmetic result having the second data width, which is smaller than the first data width, in the mantissa of the floating point format, whereby whether or not the first arithmetic result is correct can be determined with a less circuit amount.

[0015] Furthermore, since the second arithmetic result has the second data width that is smaller than the first data width, whereby an error can be securely detected from the result of the comparison if the error that is larger than a predetermined value exists in the first and the second arithmetic results.

[0016] As described above, according to the present invention, an error that has a large numerical error can be efficiently detected with less circuit amount.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0017] FIG. 1 is a schematic explaining the outline of a floating point arithmetic circuit according to a preferred embodiment of the present invention;

[0018] FIG. 2 is a schematic exemplifying a specific configuration of a first arithmetic unit according to a first preferred embodiment of the present invention;

[0019] FIG. 3 is a schematic exemplifying a specific configuration of a second arithmetic unit and a comparison circuit according to the first preferred embodiment of the present invention;

[0020] FIG. 4A is a schematic explaining a comparison that allows an error of 1 LSH, and is made by a near match detector according to the first preferred embodiment of the present invention;

[0021] FIG. 4B is a schematic explaining a comparison that allows an error of 1 LSH, and is made by a near match detector according to the first preferred embodiment of the present invention;

[0022] FIG. 5 is a schematic exemplifying a configuration of an input unit of the comparison circuit according to the first preferred embodiment of the present invention;

US 2008/0155004 A1

Jun. 26, 2008

2

[0023] FIG. 6A is a schematic exemplifying a configuration of the input unit of the comparison circuit according to the first preferred embodiment of the present invention;

[0024] FIG. 6B is a schematic exemplifying a configuration of the input unit of the comparison circuit according to the first preferred embodiment of the present invention;

[0025] FIG. 7 is a schematic exemplifying a specific configuration of a first arithmetic unit according to a second preferred embodiment of the present invention;

[0026] FIG. 8 is a schematic exemplifying a specific configuration of a second arithmetic unit and a comparison circuit according to the second preferred embodiment of the present invention;

[0027] FIG. 9 is a schematic explaining an example in a case where the preferred embodiment is applied to a iterative convergent calculation; and

[0028] FIG. 10 is a block diagram exemplifying a configuration required for the reexecution of a floating point arithmetic circuit according to the preferred embodiment of the present invention.

#### DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0029] Preferred embodiments according to the present invention are hereinafter described with reference to FIGS. 1 to 10.

[0030] FIG. 1 is a schematic explaining the outline of a floating point arithmetic circuit 100 according to a preferred embodiment of the present invention.

[0031] The floating point arithmetic circuit 100 shown in FIG. 1 comprises a first arithmetic unit 101 for outputting a first arithmetic result, a second arithmetic unit 102 for outputting a second arithmetic result, and a comparison circuit 103 for making a comparison between the first and the second arithmetic results by a predetermined bit width.

[0032] The first arithmetic unit 101 outputs the first arithmetic result when an operand, which is represented in a floating point format, is input. The first arithmetic result is data in the floating point format, and its mantissa has a first data width (such as a 52-bit width). Hereinafter, a sign, an exponent, and a mantissa, which configure the first arithmetic result, are hereinafter referred to as a first sign, a first exponent, and a first mantissa respectively. However, assume that a round off process is not executed for the first sign, exponent, and mantissa.

[0033] To the second arithmetic unit 102, the same operand as that of the first arithmetic unit 101 is input. Then, the second arithmetic unit 102 outputs the second arithmetic result. Also the second arithmetic result is data in the floating point format, and its mantissa has a second data width. Here, assume that the second data width is a data width (such as a 4-bit width) which is smaller than the first data width. Accordingly, the circuit amount of the second arithmetic unit 102 is smaller than that of the first arithmetic unit 101.

[0034] The sign, the exponent, and the mantissa, which configure the second arithmetic result, are hereinafter referred to as a second sign, a second exponent, and a second mantissa respectively.

[0035] The comparison circuit 103 makes a comparison between the first and the second arithmetic results by the second data width, and outputs a comparison result.

[0036] A double-precision floating point arithmetic circuit is described below as a specific example of the floating point arithmetic circuit 100 according to the preferred embodiment

of the present invention. However, the present invention is not limited to a double-precision floating point arithmetic circuit. As a matter of course, the present invention is also applicable, for example, to a single-precision floating point arithmetic circuit, etc.

[0037] The following description assumes that input data to the floating point arithmetic circuit 100 according to this preferred embodiment is data in a double-precision floating point format based on IEEE (Institute of Electrical and Electronic Engineers) 754. Also assume that the first operand composed of a sign S1 (1-bit width), an exponent E1 (11-bit width), and a mantissa F1 (52-bit width), and the second operand composed of a sign S2 (1-bit width), an exponent E2 (11-bit width), and a mantissa F2 (52-bit width) are used as input data.

[0038] Accordingly, also the output of the floating point arithmetic circuit 100 (namely, the first arithmetic unit 101) according to this preferred embodiment is data in the double-precision floating point format based on IEEE 754, and assumed to be output data composed of a sign SIGN (1 bit), an exponent EXP (11-bit width), and a mantissa FRAC (52-bit width).

#### First Preferred Embodiment

[0039] FIG. 2 is a schematic exemplifying a specific configuration of the first arithmetic unit 101 according to the first preferred embodiment of the present invention.

[0040] The first arithmetic unit 101 shown in FIG. 2 is a double-precision floating point adder that comprises an adder 201 for obtaining the sum of the mantissas F1 and F2, a subtractor 202 for obtaining a difference between the exponents E1 and E2, a converter 203 for converting the difference between the exponents E1 and E2 into an absolute value, a right-shifter 204 for shifting the mantissa according to the absolute value, switchers 205 to 207 for switching the output according to the positivity/negativity of the difference between the exponents E1 and E2, an exclusive OR unit 208 for obtaining the exclusive OR of the signs S1 and S2, a converter 209 for converting the output value of the adder 201 into an absolute value, a bit position detector 210 for detecting the position of a bit initially having a value 1 viewed from the most significant bit, a bit position aligner 211 for making a left-shift so that the detected bit position becomes the most significant, a round off processor 212 for executing a round off process, a decoder 213 for determining a sign SIGN from the signs S1 and S2, and the signs of the output values of the adder 201 and the subtractor 202, and a subtractor 214 for obtaining a difference between the exponent E1 or E2 and the bit position detector 210 and the bit position aligner 211.

[0041] For an addition process, the digits of the exponents E1 and E2 must be aligned. Accordingly, the subtractor 202 obtains a difference between the exponents E1 and E2. Then, the right-shifter 203 right-shifts the mantissa F2 according to the resultant difference.

[0042] Here, if the difference between the exponents E1 and E2 becomes a negative value, the mantissa F1 must be right-shifted. Therefore, an input (the mantissa F1 or F2) to the right-shifter 203 and the adder 201 is switched by switching the outputs of the switchers 205 and 206 according to the positivity/negativity of the result of the subtractor 202.

[0043] The adder 201 is an adder for adding data of a 53-bit width after restoring the most significant bit "1" in the mantissas (52 bits) of the two operands, and for outputting data of a 54-bit width. The adder 201 makes an addition or a subtract-

US 2008/0155004 A1

Jun. 26, 2008

3

tion according to the output value of the exclusive OR unit 208. The adder 201 makes an addition if the signs S1 and S2 are the same, or makes a subtraction if they differ.

[0044] The mantissas F1 and F2 are positive values in all cases. Therefore, the converter 209 takes the absolute value of the output value of the adder 201 to make the output value positive. The bit position detector 210 detects the position of a bit which initially has a value "1" viewed from the most significant bit of the addition result that becomes positive, and outputs the detected position to the bit position aligner 211.

[0045] The bit position aligner 211 left-shifts the addition result input from the converter 209 to shift the bit position detected by the bit position detector 210 to the position of the most significant bit. The bit position detected by the bit position detector 210 is input also to the subtractor 214 in order to reflect the amount of this left-shift on the exponent EXP.

[0046] The round off processor 212 executes the round off process for the value output from the bit position aligner 211. Because this round off process is a round off process based on IEEE 754, its details are omitted.

[0047] The subtractor 214 aligns the digits by subtracting the output value of the bit position detector 210 and the amount of shift made by the round off processor 212 from the exponent (E1 or E2).

[0048] The decoder 213 calculates a sign SIGN from the signs (S1, S2), the sign of the output value of the subtractor 202, and the sign of the output value of the adder 201.

[0049] With the above described process, an addition result (sign SIGN, exponent EXP, mantissa FRAC) of the first operand and the second operand is obtained. Additionally, in this preferred embodiment, the output values of the decoder 213, the switcher 207, and the converter 209 are used respectively as the first sign, the first exponent, and the first mantissa.

[0050] FIG. 3 is a schematic exemplifying a specific configuration of the second arithmetic unit 102 and the comparison circuit 103 according to the first preferred embodiment of the present invention.

[0051] To the second arithmetic unit 102, the first and the second operands are input similar to the first arithmetic unit 101. However, only high-order 4 bits are used for the mantissas. In the explanation about FIG. 3, the first operand composed of the sign S1 (1-bit width), the exponent E1 (11-bit width), and the mantissa F1 (4-bit width), and the second operand composed of the sign S2 (1-bit width), the exponent E2 (11-bit width), and the mantissa F2 (4-bit width) are used as input data.

[0052] The second arithmetic unit 102 shown in FIG. 3 is a low-precision floating point adder comprising an adder 301 for obtaining the sum of the mantissas F1 and F2, a subtractor 202 for obtaining a difference between the exponents E1 and E2, a converter 203 for converting the difference between the exponents E1 and E2 into an absolute value, a right-shifter 302 for right-shifting the mantissa according to the absolute value, switchers 205 to 207 for switching the output according to the positivity/negativity of the difference between the exponents E1 and E2, an exclusive OR unit 208 for obtaining the exclusive OR of the signs S1 and S2, and a converter 303 for converting the output value of the adder 201 into an absolute value.

[0053] Similar to the first arithmetic unit 101, the right-shifter 302 right-shifts the mantissa F2 according to the difference between the exponents E1 and E2, which is calculated by the subtractor 202. Additionally, the right-shifter 302 switches an input (the mantissa F1 or F2) to the adder 301 and

the right-shifter 302 by switching the outputs of the switchers 205 and 206 according to the positivity/negativity of the output value of the subtractor 202.

[0054] The adder 301 is, for example, an adder for outputting data of a 5-bit width. The adder 301 makes an addition or a subtraction according to the signal of the exclusive OR unit 208 similar to the first arithmetic unit. The converter 303 takes the absolute value of the output value of the adder 301 to make the output value positive.

[0055] The decoder 213 calculates the second sign from the signs S1 and S2, and the signs of the output values of the subtractor 202 and the adder 201.

[0056] With the above described process, the addition of the first and the second operands is made to calculate the second arithmetic result. In this preferred embodiment, the output values of the decoder 213, the switcher 207, and the converter 303 are used respectively as the second sign, the second exponent, and the second mantissa.

[0057] The comparison circuit 103 shown in FIG. 3 comprises a near match detector 304 for determining whether or not an error is within a predetermined range by making a comparison between the first and the second mantissas, a mismatch detector 305 for detecting a mismatch by making a comparison between the first and the second exponents, a mismatch detector 306 for detecting a mismatch by making a comparison between the first and the second signs, a logical OR unit 307 for detecting an error by obtaining the logical OR of the output values of the near match detector 304, and the mismatch detectors 305 and 306.

[0058] The first and the second mantissas are input to the near match detector 304. Here, the mantissa in the second arithmetic result inevitably includes an error of 1 least significant bit (LSB: Least Significant Bit). Therefore, only 1LSB may sometimes differ even if the calculation is made with proper precision. An error at this time is hereinafter referred to as an "error of 1LSB".

[0059] For this reason, not a perfect match detection circuit but the near match detector 304 for making a comparison that allows the error margin of 1 LSB is used in the comparison circuit 103 according to this preferred embodiment. If the error margin is made large, for example, if an error margin of 2 least significant bits is allowed, also a comparison that allows the error margin of 2LSB can be made by making a comparison between high-order bits excluding the LSB, which are obtained by subtracting 1 bit from the input data of the near match detector 304. Its details will be described later with reference to FIG. 5.

[0060] The first and the second mantissas are input to the near match detector 304. The near match detector 304 determines whether or not the error of both of the mantissas is within a predetermined range. For example, the mantissas are determined to match if the error is within the predetermined range (the range of the error of 1LSB), or the mantissas are determined to mismatch if the error is not within the predetermined range (the range of the error of 1LSB). A specific configuration example will be described with reference to FIG. 6A and 6B.

[0061] The first and the second exponents are input to the mismatch detector 305. The mismatch detector 305 determines whether the exponents either match or mismatch by calculating a bitwise exclusive OR, and by performing a logical OR operation for the outputs. Similarly, the first and the second signs are input to the mismatch detector 306. Then,



US 2008/0155004 A1

Jun. 26, 2008

4

the mismatch detector 306 determines whether the signs either match or mismatch by performing an exclusive OR operation.

[0062] The outputs of the near match detector 304, and the mismatch detectors 305 and 306 are input to the logical OR unit 307. For example, if the near match detector 304, and the mismatch detectors 305 and 306 respectively output 1 upon detection of a mismatch, the logical OR unit 307 outputs 1 when at least one or more of the near match detector 304, and the mismatch detectors 305 and 306 detect a mismatch. As a result, a mismatch between the first and the second arithmetic results, namely, a numerically significant error of the first arithmetic unit 101 can be detected.

[0063] FIG. 4A and 4B are a schematic explaining the comparison that allows the error of 1LSB, which is made by the near match detector 304 according to the first preferred embodiment of the present invention.

[0064] FIG. 4A shows a case where A and B, which are input to the near match detector 304, are identical other than LSB. This is, for example, a case where A and B are "11110" and "11111" respectively. If the exclusive OR of A and B is calculated, its arithmetic result C becomes a sequence of "0" other than the LSB, and only the LSB becomes "1".

[0065] FIG. 4B shows a case where A and B, which are input to the near match detector 304, are different by 1LSB. This is, for example, a case where the numeric values of A and B are different only 1LSB such as a case where A and B are "10000" and "01111" respectively. If the exclusive OR of A and B is calculated, its arithmetic result C becomes "0" in a bit position where A and B match (a sequence of "0"), and becomes "1" in a bit position where A and B do not match (a sequence of "1").

[0066] It is proved from the cases shown in FIGS. 4A and 4B that the arithmetic result C obtained by calculating the exclusive OR becomes a bit pattern where "0" is consecutive sequentially from the most significant bit, and "1" is consecutive after "1" once "1" appears. Namely, if A and B are the above described bit patterns, it can be determined that A and B are different by 1LSB or less.

[0067] If the arithmetic result C is a bit pattern where all of bits are "1", this is a case where an error of A and B is the error of 1LSB. Or, if the arithmetic result C is a bit pattern where all of bits are "0", A and B perfectly match. Therefore, this is regarded as a case where A and B are different by 1LSB or less.

[0068] A specific configuration example of the comparison circuit 103 is described below with reference to FIGS. 5 and 6B.

[0069] FIG. 5 is a schematic exemplifying a configuration of an input unit of the comparison circuit 103 according to the first preferred embodiment of the present invention. A0 to An represent the bits of the first mantissa when a bit width is n, whereas B0 to Bn represent the bits of the second mantissa when a bit width is n. In the case of this preferred embodiment, it is sufficient to set n to 4.

[0070] The input unit of the comparison circuit 103 comprises exclusive OR circuits 500 to 50n, which calculate the exclusive OR of each of A0 to An and each of B0 to Bn, and output arithmetic results X0 to Xn. Assume that the arithmetic results of the exclusive OR of A0 and B0, A1 and B1, ..., An and Bn are respectively X0, X1, ..., Xn if the comparison circuit 103 is implemented as a comparison circuit that allows the error of 1LSB, or assume that the arithmetic results of the exclusive OR of the A1 and B1, A2 and B2, ..., An and Bn are

respectively X0, X1, ..., Xn-1, and "0" is Xn if the comparison circuit 103 is implemented as a comparison circuit that allows the error of 2LSB, as shown in FIG. 5.

[0071] FIG. 6A and 6B is a schematic exemplifying a configuration of a determining unit of the comparison circuit 103 according to the first preferred embodiment of the present invention. For simplicity of explanation, a comparison circuit of an 8-bit precision (n=7) is described. However, the comparison circuit is not limited to this configuration as a matter of course.

[0072] The arithmetic results X0 and X1, X2 and X3, X4 and X5, and X6 and X7 are respectively input to logical AND units 601 to 604, and logical OR units 611 to 614.

[0073] Outputs O10 and O32 of the logical AND units 601 and 602 are input to a logical AND unit 605, which then makes an output O30. Similarly, outputs O54 and O76 of the logical AND units 603 and 604 are input to a logical AND unit 606, which then makes an output O74.

[0074] Accordingly, an output Omm indicates that the m-th bit to the n-th bit are "1".

[0075] Additionally, outputs Z10 and Z32 of the logical OR units 611 and 612 are input to a logical AND unit 615, which then makes an output Z30. Similarly, outputs Z54 and Z76 of the logical OR units 613 and 614 are input to a logical AND unit 616, which then makes an output Z74.

[0076] Accordingly, an output Zmm indicates that the m-th bit to the n-th bit are "0". With the above described process, the sequence of "0" and that of "1" in X0 to X7 are obtained.

[0077] Furthermore, the outputs Z30 and Z74 are input to a logical AND unit 621, Z32, Z74 and X1 are input to a logical AND unit 622, O10, Z74 and X2 are input to a logical AND unit 623, O30, Z76 and the inversion of X5 are input to a logical AND unit 624, O30, O54 and the inversion of X7 are input to a logical AND unit 625, and O30 and O74 are input to a logical AND unit 626. The outputs of the logical AND units 621 to 626 are input to a logical AND unit 635 via inverters 629 to 634.

[0078] In the above described configuration, for example, if (1) O30=O74=1, it can be determined that all of X0 to X7 are 1. Or, if (2) X7=0, and O54=O30=1, (3) Z76=X5=0, and O30=1, (4) Z74=0, and X2=O10=1, or (5) Z74=Z32=0, and X1=1, it can be determined that 0 is consecutive from the most significant bit to a predetermined bit and then 1 is consecutive in X0 to X7. Additionally, if (6) Z74=Z30=0, it can be determined that all of X1 to X7 are 0 (A and B perfectly match).

[0079] In the above described cases (1) to (6), the logical AND unit 635 outputs "1". Namely, the logical AND unit 635 outputs "1" if the error of A and B is larger than 1LSB.

[0080] As a result, whether or not an error of the first and the second mantissas, which are input to the comparison circuit 103, is a near match of the range of 1LSB can be determined from the output (0 or 1) of the comparison circuit 103.

#### Second Preferred Embodiment

[0081] FIG. 7 is a schematic exemplifying a specific configuration of the first arithmetic unit 101 according to the second preferred embodiment of the present invention.

[0082] The first arithmetic unit 101 shown in FIG. 7 is a double-precision floating point multiplier comprising a multiplier 701 for obtaining the product of the mantissas P1 and P2, a round off processor 712 for executing a round off process, an adder 702 for calculating an exponent from the

US 2008/0155004 A1

Jun. 26, 2008

5

sum of exponents E1 and E2, and an exclusive OR unit 703 for calculating a sign from the exclusive OR of the signs S1 and S2.

[0083] The multiplier 701 restores the most significant bit "1" in the mantissas (52 bits) of the two operands, and multiplies data of a 53-bit width. Additionally, the adder 702 adds the exponents of the two operands. The exclusive OR unit 703 obtains the exclusive OR of the signs of the two operands.

[0084] The round off process conforming to IEEE standards is executed by the round off processor 212 for the result of the multiplication made by the multiplier 701. If a digit carry occurs as a result, the adder 704 makes a correction.

[0085] With the above described process, a multiplication result (the sign SIGN, the exponent EXP, and the mantissa FRAC) of the first and the second operands is obtained. In this preferred embodiment, the output values of the exclusive OR unit 703, the adder 702, and the multiplier 701 are used respectively as the first sign, the first exponent, and the first mantissa.

[0086] FIG. 8 is a schematic exemplifying a specific configuration of the second arithmetic unit 102 and the comparison circuit 103 according to the second preferred embodiment of the present invention.

[0087] In a similar manner as in the first preferred embodiment, the first and the second operands are input to the second arithmetic unit 102. However, only high-order 4 bits are used for the mantissa. Explanation about FIG. 8 assumes that the first operand composed of a sign S1 (1-bit width), an exponent E1 (11-bit width), and a mantissa I1 (4-bit width), and the second operand composed of a sign S2 (1-bit width), an exponent E2 (11-bit width), and a mantissa I2 (4-bit width) are used as input data.

[0088] The second arithmetic unit 102 shown in FIG. 8 is a low-precision floating point multiplier comprising a multiplier 801 for obtaining the product of the mantissas I1 and I2, an adder 702 for calculating the second exponent from the sum of the exponents E1 and E2, and an exclusive OR unit 804 for calculating the first sign from the exclusive OR of the signs S1 and S2.

[0089] The multiplier 801 is, for example, a multiplier for making a multiplication of 4+1 (the bit width of the second mantissa+1) bits. The multiplier 801 restores the most significant bit "1" in the mantissas (4 bits) of the two operands, and multiplies data of a 5-bit width.

[0090] The adder 702 adds the exponents of the two operands and (+1023). The exclusive OR unit 804 obtains the exclusive OR of the signs of the two operands and the first sign.

[0091] With the above described process, the multiplication of the first and the second operands is made to calculate the second arithmetic result. In this preferred embodiment, the output values of the exclusive OR unit 804, the adder 702, and the multiplier 801 are used respectively as the second sign, the second exponent, and the second mantissa.

[0092] The comparison circuit 103 shown in FIG. 8 comprises a near match detector 802 for determining whether or not an error is within a predetermined range (the range of the error of 2L/3B) by making a comparison between the first and the second mantissas, a mismatch detector 803 for detecting a mismatch by making a comparison between the first and the second exponents, a mismatch detector 804 for detecting a mismatch by making a comparison between the first and the second signs, and a logical OR unit 805 for detecting an error by obtaining the logical OR of the output results of the near

match detector 802, and the mismatch detectors 803 and 804. Although the exclusive OR unit 804 used in the second arithmetic unit 102 is shared as the mismatch detector 804 used in the comparison circuit 103, they may be configured separately.

[0093] The first and the second mantissas are input to the near match detector 802. The near match detector 802 determines whether or not an error of the first and the second mantissas is within a predetermined range (the range of the error of 2L/3B).

[0094] Here, the following equation is calculated as a multiplication made, for example, in a case where errors ( $\Delta a, \Delta b$ ) are included respectively in the first operand a and the second operand b.

$$a^*(1-\Delta a)^{1/2}b^*(1-\Delta b)^{1/2} \approx a^*b^*(1-\Delta a-\Delta b+\Delta a^2/2)$$

[0095] Here, assuming that the first and the second mantissas are 4 bits,  $\Delta a$  and  $\Delta b$  become a value smaller than 1/32. However, the error ( $\Delta a+\Delta b-\Delta a^2/2$ ) of the multiplication result does not become a value smaller than 1/32 like the error of the addition result although it becomes a value smaller than 1/16. Accordingly, a near match detector for making a comparison that allows the error of 2L/3B is used as the near match detector 802 according to this preferred embodiment. Since the specific configuration example was described with reference to FIGS. 5 and 6, it is omitted here.

[0096] The first and the second exponents are input to the mismatch detector 803. The mismatch detector 803 determines whether both of the exponents either match or mismatch by calculating a bitwise exclusive OR, and by performing a logical OR operation for the results of the exclusive OR. The first sign, and the signs S1 and S2 of the first and the second operands are input to the exclusive OR unit 804, which then obtains the exclusive OR of the signs.

[0097] The logical OR unit 805 calculates the logical OR of the near match detector 802, and the mismatch detectors 803 and 804. Accordingly, for example, if the near match detector 802 and the mismatch detectors 803 and 804 respectively output 1 upon detection of a mismatch, the logical OR unit 805 outputs 1 when at least one or more of the nearly match detector 802, and the mismatch detectors 803 and 804 detect a mismatch. As a result, a mismatch between the first and the second arithmetic results, namely, a numerically significant error of the first arithmetic unit 101 can be detected.

[0098] For the first and the second arithmetic units, two sets of sign and exponent calculation circuits of the same scale are required. The circuit amount of the mantissa of the second arithmetic unit is significantly reduced as will be described next.

[0099] Normally, the circuit amount of an adder or a shifter is approximately proportional as follows if a bit width is assumed to be N.

$$N^2 \log_2 N$$

[0100] Accordingly, for example, a circuit amount required for the adder 301 or the right-shifter 302 of 5 bits is approximately 1/20 in comparison with the adder 201 or the right-shifter 204 of 52 bits.

[0101] Therefore, in the first preferred embodiment, the second arithmetic unit 102 and the comparison circuit 103 can be implemented with a circuit amount of 10 percent or less of the entire floating point arithmetic circuit 100 if the mantissa is 52 bits.

[0102] Additionally, the circuit amount of the multiplier is approximately proportional to the square of N. Accordingly,

US 2008/0155004 A1

Jun. 26, 2008

6

the multiplier 801 of 5 bits can be implemented, for example, with a circuit amount of approximately 1/100 in comparison with the multiplier 701 of 53 bits.

[0103] Therefore, in the second preferred embodiment, the second arithmetic unit 102 and the comparison circuit 103 can be implemented with a circuit amount of 2 to 3 percent of the entire floating point arithmetic circuit 100.

[0104] Note that the mantissa is equal to or larger than 1.0 and smaller than 2.0 if omitted 1.0 is complemented. A comparison is made only for the first and the second signs, and the first and the second exponents by regaining the value of the mantissa as 1.0, whereby required hardware can be further reduced since, for example, the adder 301 and the multiplier 801 become unnecessary although a precision for detecting an error deteriorates.

[0105] FIG. 9 is a schematic explaining an example when the preferred embodiment is applied to a iterative convergent calculation.

[0106] The graph represented in FIG. 9 is a graph of a second degree equation  $Y=X^2X-3^*X+2$ . A case where a solution to this second degree equation is obtained with Newton Lapson method, which is an iterative convergent calculation, is described below.

[0107] For the second degree equation  $Y(X)=X^2X-3^*X+2$ ,  $Y=0$  if  $X=1$  and  $X=2$ . Obtainment of the solution  $X=1$  is considered starting from  $X=0$ .

[0108] Since  $dY/dX=2^*X-3$ , the gradient of  $Y$  when  $X=0$  is  $Y'(0)=-3$ , and  $Y(0)=2$ . If the next value of  $X$  is obtained from  $X=X-Y(0)/Y'(0)$ ,  $X=2/3$ . Similarly, the gradient when  $X=2/3$  is  $Y'(2/3)=-5/3$ , and  $Y(2/3)=4/9$ . Since the next value of  $X=14/15$ ,  $Y'(14/15)=-17/15$ , and  $Y(14/15)=16/225$  in a similar manner. The further next value of  $X$  is  $X=254/225$ . At this time, an error from the correct solution  $X=1$  is reduced to approximately 0.4 percent.

[0109] As is evident from the above calculation, even if an error occurs in the next  $X$  due to an occurrence of an error in the calculation of a correction amount of  $Y$ ,  $Y'$  or  $X$ ,  $X$  converges to 1.0 with iterations as far as  $X$  remains to be smaller than 1.5.

[0110] If an error occurs due to a calculation error, and goes away from the value of a solution, the number of iterations increases and an additional calculation time is required. However, since the occurrence frequency of the above described error caused by radiation particles, etc. is low, an influence exerted on the entire calculation time is on an ignorable order.

[0111] In the meantime, if an error due to a calculation error is large, and  $X$  becomes larger than 1.5, the value of  $X$  converges to 2.0 with subsequent iterations. Because this is also a solution to  $X^2X-3^*X+2=0$ , this value is correct as a solution to this second degree equation. However, in an actual iterative convergent calculation, if a calculation error at midpoint is large,  $X$  converges to an unintended point, or a convergent point cannot be found in a high degree curve and  $X$  diverges in some cases.

[0112] For example, the first principle calculation, which is used in a main- or bio-field mainly utilized by a supercomputer, etc., is made to converge the energy of an entire system to a minimum with iterations such that the position of an atom is moved by calculating repulsive force between electrons, and by totaling the repulsive forces, and force exerted on each atom is calculated.

[0113] The force exerted on each atom is the sum of forces from electrons of all of other atoms. Even if some error such as an error caused by an intermittent error of hardware occurs

in the calculation of force exerted from one electron, the electron is one of several thousands or several tens of thousands of electrons. Therefore, its influence is reduced.

[0114] As described above, with a normal calculation, the influence of an error, which is small as a value, is automatically modified during the course of a tentative convergent process in most cases as far as an error that causes a significant difference of digits does not occur in an individual calculation.

[0115] Accordingly, reexecution is made by detecting an error, which causes an incomparable large error, by using the floating point arithmetic circuit 100 according to this preferred embodiment, whereby an intermittent error caused by a failure, which is not a fixed failure such as the collision of radiation particles caused by an alpha ray or a cosmic ray, can be corrected.

[0116] FIG. 10 is a block diagram exemplifying a configuration required for the reexecution of the floating point arithmetic circuit 100 according to the preferred embodiment of the present invention. For the reexecution when a hardware error is detected, its details are described in the following documents. Therefore, its outline is described.

[0117] 1. H. Ando, T. Kitamura, M. Shebanow, M. Butler, U.S. Pat. No. 6,519,730, "Computer and error recovery method for the same" 2. H. Ando, Y. Yoshida, A. Inoue, I. Sugiyama, T. Asakawa, K. Morita, T. Muta, T. Motokurumada, S. Okada, H. Yamashita, Y. Sawakawa, A. Konno, R. Yamashita, H. Sugiyama, "A 1.3-GHz fifth-generation SPARC64 microprocessor", Solid-State Circuits, IEEE Journal, Volume 38, Nov. 11, 2003, 1896-1905.

[0118] The floating point arithmetic circuit 1000 shown in FIG. 10 comprises at least a processor state 1001 for holding a state required for the execution of a program, a checkpoint array 1002 for storing a state when the processor operation is properly performed without error in certain intervals, a commit pointer 1003 pointing to an entry that stores a state where the execution of an instruction is properly completed last, a current pointer 1004 pointing to an entry that stores an immediately preceding state, a backup control 1005 for performing a control to resume the execution from a state by extracting the state pointed to by the commit pointer from the checkpoint array 1002, and by restoring the state in the processor state 1001 when a misoperation, etc. occurs, a branch unit 1006 for handling a branch instruction, an error detection unit 1007 for monitoring a misoperation of each unit, a control unit 1008 for controlling each unit, an arithmetic execution unit 1009 configured with the first arithmetic unit 101, and an error check unit 1010 for monitoring the correctness/incorrectness of the arithmetic result of the arithmetic execution unit 1009 by being configured with the second arithmetic unit 102 and the comparison circuit 103.

[0119] If the execution is speculatively made by predicting a direction before a branch destination is determined, the state (contents of the processor state 1001) immediately preceding the execution of the branch instruction is stored in an entry of the checkpoint array 1002 pointed to by the current pointer 1004. Then, the execution of the instruction at the predicted branch destination is started. Then, the branch unit 1006 increments the current pointer 1004 by 1. Additionally, the branch unit 1006 performs a control to make the commit pointer 1003 always point to the completion state of the last execution of an instruction by monitoring the completion of the execution of the instruction, and by incrementing the



US 2008/0155004 A1

Jun. 26, 2008

7

commit pointer 1003 by 1 upon completion of the execution up to an entry next to the entry pointed to by the commit pointer 1003.

[0120] If a branch prediction miss is found, the error of the branch prediction is modified by extracting a checkpoint immediately preceding a conditional branch instruction from the checkpoint array 1002, and by restoring the processor state to resume the execution from the instruction immediately succeeding the checkpoint.

[0121] When the error detection unit 1007 detects an error with the processes referred to in the first and the second preferred embodiments, the error detection unit 1007 notifies the backup control 1005 of the misoperation.

[0122] Upon receipt of the misoperation notification, the backup control 1005 extracts the state in which the execution is properly completed and which is pointed to by the commit pointer 1003, and restores the extracted state in the processor state 1001. Then, the execution is resumed from this state (reexecution is made). As a result, an intermittent error caused by the collision of radiation particles, etc. can be corrected. This procedure is similar to that for correcting a branch prediction error, and can effectively use the circuit existing for a branch prediction miss handling.

[0123] As described above, the second arithmetic unit 102 and the comparison circuit 103 of a short precision (such as 4 bits) are added to the first arithmetic unit 101 in order for an error check, and a comparison that allows a predetermined error margin (such as an error of 11.5B) of the first and the second arithmetic results is made, whereby an error which causes a large numerical error can be efficiently detected. Additionally, the detected error can be corrected by using the above described reexecution mechanism.

[0124] Furthermore, since the second mantissa according to this preferred embodiment has a 4-bit width, the second arithmetic unit 102 and the comparison circuit 103 can be implemented with a circuit amount of approximately 2 to 3 percent of the entire floating point arithmetic circuit 100 in the case of the floating point multiplier, or with a circuit amount of approximately 10 percent or less of the entire floating point arithmetic circuit 100 in the case of the floating point adder. Namely, a circuit amount required to detect an error can be reduced in comparison with conventional techniques.

[0125] Still further, even for an intermittent error, which causes a significant difference of digits of the numerical value of an arithmetic result and cannot be detected with a conventional method for detecting an error of a principal portion of a floating point arithmetic unit with a parity check or a residue check, an error larger than the precision of the second mantissa can be securely detected with the method according to the present invention.

[0126] Still further, when an iterative convergent calculation is made, an error that is smaller than an allowed numerical error is corrected during the iterative convergent calculation, whereby an added circuit the amount of which is less than a conventional error detection method can be implemented, and the probability of occurrence of a problem such as a failure of convergence of the calculation, or an error occurrence in an arithmetic result despite being terminated without detecting an abnormality can be significantly reduced.

What is claimed is:

1. A floating point arithmetic circuit, comprising:
  - a first arithmetic unit, to which an operand represented in a floating point format is input, for outputting a first arithmetic result having a first data width in a mantissa of the floating point format;
  - a second arithmetic unit, to which the operand is input, for outputting a second arithmetic result having a second data width, which is smaller than the first data width, in a mantissa of the floating point format; and
  - a comparison circuit for making a comparison between the mantissas of the first and the second arithmetic results by the second data width from each predetermined bit.
2. The floating point arithmetic circuit according to claim 1, wherein:
  - said comparison circuit comprises
    - an exclusive OR circuit for performing a bitwise exclusive OR operation of the mantissas of the first and the second arithmetic results by the second data width from a most significant bit; and
    - a mantissa comparison circuit for determining whether or not an arithmetic result of the exclusive OR operation is a predetermined bit pattern, and for outputting a comparison result indicating a match if the arithmetic result is the predetermined bit pattern; and
  - the first and the second arithmetic results are determined to match if said mantissa comparison circuit outputs the comparison result indicating the match.
3. The floating point arithmetic circuit according to claim 2, wherein:
  - the predetermined bit pattern is any of a bit pattern where all of bits are 1, a bit pattern where all of bits are 0, and a bit pattern where 0 is consecutive from the most significant bit to an arbitrary bit and 1 is consecutive from a bit next to the arbitrary bit to a predetermined bit.
4. The floating point arithmetic circuit according to claim 2, further comprising:
  - a sign comparison circuit for making a comparison between signs of the first and the second arithmetic results, and for outputting a comparison result indicating a match if the signs match as a result of the comparison, wherein
  - the first and the second arithmetic results are determined to match if said mantissa comparison circuit and said sign comparison circuit output the comparison result indicating the match.
5. The floating point arithmetic circuit according to claim 2, further comprising:
  - an exponent comparison circuit for making a comparison between exponents of the first and the second arithmetic results, and for outputting a comparison result indicating a match if the exponents match as a result of the comparison, wherein
  - the first and the second arithmetic results are determined to match if said mantissa comparison circuit and said exponent comparison circuit output the comparison result indicating the match.
6. The floating point arithmetic circuit according to claim 1, further comprising:
  - a reexecution circuit for causing said first arithmetic unit to reexecute an arithmetic according to the result of the comparison made by said comparison circuit.
7. An arithmetic examination circuit for examining a first arithmetic result, which is connected to a first arithmetic unit,



US 2008/0155004 A1

Jun. 26, 2008

8

to which an operand represented in a floating point format is input, for outputting the first arithmetic result having a first data width in a mantissa of the floating point format, comprising:

- a second arithmetic unit, to which the operand is input, for outputting a second arithmetic result having a second data width, which is smaller than the first data width, in a mantissa of the floating point format; and
- a comparison circuit for making a comparison between the mantissas of the first and the second arithmetic results by the second data width from each predetermined bit.

8. The arithmetic examination circuit according to claim 7, wherein:

said comparison circuit comprises

- an exclusive OR circuit for performing a bitwise exclusive OR operation of the mantissas of the first and the second arithmetic results by the second data width from a most significant bit, and
- a mantissa comparison circuit for determining whether or not an arithmetic result of the exclusive OR operation is a predetermined bit pattern, and for outputting a comparison result indicating a match if the arithmetic result is the predetermined bit pattern; and

the first and the second arithmetic results are determined to match if said mantissa comparison circuit outputs the comparison result indicating the match.

9. The arithmetic examination circuit according to claim 8, wherein

the predetermined bit pattern is any of a bit pattern where all of bits are 1, a bit pattern where all of bits are 0, and a bit pattern where 0 is consecutive from the most significant bit to an arbitrary bit and 1 is consecutive from a bit next to the arbitrary bit to a predetermined bit.

10. The arithmetic examination circuit according to claim 8, further comprising

- a sign comparison circuit for making a comparison between signs of the first and the second arithmetic results, and for outputting a comparison result indicating a match if the signs match as a result of the comparison, wherein

the first and the second arithmetic results are determined to match if said mantissa comparison circuit and said sign comparison circuit output the comparison result indicating the match.

11. The arithmetic examination circuit according to claim 8, further comprising

- an exponent comparison circuit for making a comparison between exponents of the first and the second arithmetic results, and for outputting a comparison result indicating a match if the exponents match as a result of the comparison, wherein

the first and the second arithmetic results are determined to match if said mantissa comparison circuit and said exponent comparison circuit output the comparison result indicating the match.

12. The arithmetic examination circuit according to claim 7, further comprising

- a reexecution circuit for causing said first arithmetic unit to reexecute an arithmetic according to the result of the comparison made by said comparison circuit.

13. An information processing device, comprising:

a first arithmetic unit, to which an operand represented in a floating point format is input, for outputting a first arithmetic result having a first data width in a mantissa of the floating point format;

a second arithmetic unit, to which the operand is input, for outputting a second arithmetic result having a second data width, which is smaller than the first data width, in a mantissa of the floating point format; and

a comparison circuit for making a comparison between the mantissas of the first and the second arithmetic results by the second data width from each predetermined bit.

14. The information processing device according to claim 13, wherein:

said comparison circuit comprises

- an exclusive OR circuit for performing a bitwise exclusive OR operation of the mantissas of the first and the second arithmetic results by the second data width from a most significant bit, and
- a mantissa comparison circuit for determining whether or not an arithmetic result of the exclusive OR operation is a predetermined bit pattern, and for outputting a comparison result indicating a match if the arithmetic result is the predetermined bit pattern; and

the first and the second arithmetic results are determined to match if said mantissa comparison circuit outputs the comparison result indicating the match.

15. The information processing device according to claim 14, wherein

the predetermined bit pattern is any of a bit pattern where all of bits are 1, a bit pattern where all of bits are 0, and a bit pattern where 0 is consecutive from the most significant bit to an arbitrary bit and 1 is consecutive from a bit next to the arbitrary bit to a predetermined bit.

16. The information processing device according to claim 14, further comprising

- a sign comparison circuit for making a comparison between signs of the first and the second arithmetic results, and for outputting a comparison result indicating a match if the signs match as a result of the comparison, wherein

the first and the second arithmetic results are determined to match if said mantissa comparison circuit and said sign comparison circuit output the comparison result indicating the match.

17. The information processing device according to claim 14, further comprising

- an exponent comparison circuit for making a comparison between exponents of the first and the second arithmetic results, and for outputting a comparison result indicating a match if the exponents match as a result of the comparison, wherein

the first and the second arithmetic results are determined to match if said mantissa comparison circuit and said exponent comparison circuit output the comparison result indicating the match.

18. The information processing device according to claim 13, further comprising

- a reexecution circuit for causing said first arithmetic unit to reexecute an arithmetic according to the result of the comparison made by said comparison circuit.

US 2008/0155004 A1

Jun. 26, 2008

9

19. An arithmetic method for use in an arithmetic circuit for examining a first arithmetic result output by a first arithmetic unit by using a second arithmetic result output by a second arithmetic unit, comprising:

inputting an operand represented in a floating point format, and outputting the first arithmetic result having a first data width in a mantissa of the floating point format;

inputting the operand, and outputting the second arithmetic result having a second data width, which is smaller than the first data width, in a mantissa of the floating point format; and

making a comparison between the mantissas of the first and the second arithmetic results by the second data width from each predetermined bit.

20. An arithmetic examination method for use in an arithmetic examination circuit for examining a first arithmetic result, which is connected to a first arithmetic unit, to which an operand represented in a floating point format is input, for outputting the first arithmetic result having a first data width in a mantissa of the floating point format, comprising:

inputting the operand, and outputting a second arithmetic result having a second data width, which is smaller than the first data width, in a mantissa of the floating point format; and

making a comparison between the mantissas of the first and the second arithmetic results by the second data width from each predetermined bit.

\* \* \* \* \*

JP 2012-530966 A 2012.12.6

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号  
特表2012-530966  
(P2012-530966A)

(43) 公表日 平成24年12月6日(2012.12.6)

(51) Int. Cl.

F I

テーマコード (参考)

GO6F 7/00 (2006.01) GO6F 7/00 202 5B013

GO6F 9/38 (2006.01) GO6F 9/38 370A

審査請求 未請求 予備審査請求 有 (全 45 頁)

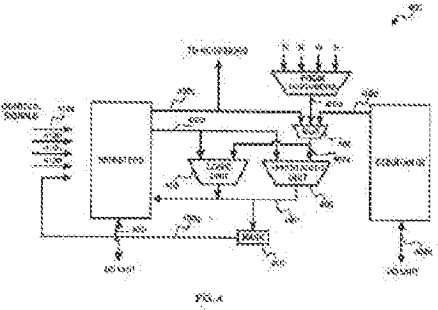
(21) 出願番号	特願2012-516227 (P2012-516227)	(71) 出願人	511308255 シンギュラー コンピューティング、エル エルシー
(86) (22) 出願日	平成22年6月16日 (2010.6.16)		アメリカ合衆国、マサチューセッツ州 O 2465、ニュートン、リージェント・ス トリート 10
(86) 翻訳文提出日	平成24年2月17日 (2012.2.17)	(74) 代理人	100121762 弁理士 杉山 直人
(86) 国際出願番号	PCT/US2010/038769	(74) 代理人	100126767 弁理士 白銀 博
(87) 国際公開番号	W02010/148054	(72) 発明者	ベイツ、ジョゼフ アメリカ合衆国、マサチューセッツ州 O 2465、ニュートン、リージェント・ス トリート 10
(87) 国際公開日	平成22年12月23日 (2010.12.23)		Fターム(参考) 5B013 DD01
(31) 優先権主張番号	61/218,691		
(32) 優先日	平成21年6月19日 (2009.6.19)		
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	12/816,201		
(32) 優先日	平成22年6月15日 (2010.6.15)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 コンパクトな演算処理要素を用いたプロセッシング

(57) 【要約】

プログラマブル及び/又は大規模並列プロセッサーあるいはその他のデバイスのような、プロセッサまたはその他のデバイスであって、低精度ハイ・ダイナミック・レンジ(LPHDR演算)の数値に対して演算処理(例えば、必須ではないが、加算、乗算、減算、そして除算の内の1つまたはそれ以上を含む処理)を実行するために設計された処理要素を備える。かかるプロセッサやその他のデバイスは、例えば、単一のチップ上で実現することができる。単一のチップ上で実現できるかどうかにかかわらず、本発明のある実施例におけるプロセッサあるいはその他のデバイスの中にあるLPHDR演算要素の数は、もし演算要素が存在するならば、演算要素の数ははるかに超える(例えば、少なくとも20+3倍の数だけ超える)ものである。ここでいう演算要素は、プロセッサやその他のデバイスの中に在って、従来の精度(32ビットあるいは64ビットの浮動小数点演算のような精度)で、ハイ・ダイナミック・レンジの演算を行うように設計された演算要素をいう。



(2)

JP 2012-530966 A 2012.12.6

## 【特許請求の範囲】

## 【請求項1】

複数の低精度ハイ・ダイナミック・レンジ(LPHDR)実行ユニットであって、複数の第2の数値を表す複数の第1の出力信号を生み出すために、複数の第1の数値を表す複数の第1の入力信号に対して複数の第1の処理を並列に実行するように構成されており、

複数のLPHDR実行ユニットの各々に関するものであって、  
LPHDR実行ユニットが実行できるように構成された少なくとも1回の処理に関して、

少なくとも1つの処理への許容できる有効な入力ダイナミック・レンジは、少なくとも1/65,000から65,000までの幅を有し、当該処理への許容できる有効な入力の少なくとも5%に関して、そして当該処理によって生み出される少なくとも1つの出力信号に関して、  
当該処理への許容できる有効な入力の少なくとも5%から同じ入力までの繰り返し実行される当該処理において、その入力に対する当該処理を実行するLPHDR演算実行ユニットの少なくとも1つの出力信号によって表される数値の統計学的平均が、その同じ入力の数値に対する処理の正確な数学的計算の結果より少なくとも0.05%だけ異なる

LPHDR演算実行ユニットと、

少なくとも1つの高精度演算実行ユニットであって、少なくとも1つの第4の数値を表す少なくとも1つの第2の出力信号を生み出すために、少なくとも1つの第3の数値を表す少なくとも1つの第2の入力信号に対する少なくとも1つの第2の演算処理を実行するように構成されており、

少なくとも1つの高精度演算実行ユニットは、32ビットまたはそれ以上のビット幅を有する浮動小数点数に対して、少なくとも1回の演算処理を実行するように構成されている高精度実行ユニットとからなり、

デバイス中のLPHDR演算実行ユニットの数は、少なくとも1つの高精度実行ユニットの数に対して、少なくとも $20+3 \times$  [高精度実行ユニットの数]だけ超えた数であることを特徴とするデバイス。

## 【請求項2】

第1の低精度ハイ・ダイナミック・レンジ(LPHDR)演算実行ユニットであって、第2の数値を表す第1の出力信号を生み出すために、第1の数値を表す第1の入力信号に対して第1の演算処理を実行するように構成されており、

第1の処理への許容できる有効な入力ダイナミック・レンジは、少なくとも1/65,000から65,000までの幅を有し、第1の処理への許容できる有効な入力の少なくとも5%に関して、第1の処理への許容できる有効な入力の少なくとも5%から各特定の入力までの繰り返し実行される第1の処理において、その入力に対する第1の処理を実行するLPHDR演算実行ユニットの第1の出力信号によって表される数値の統計学的平均が、その同じ入力の数値に対する第1の演算処理の正確な数学的計算の結果より少なくとも0.05%だけ異なる

LPHDR演算実行ユニットを備え、

32ビットまたはそれ以上のビット幅を有する浮動小数点数に対して、少なくとも1回の演算処理を実行するように構成されている高精度演算実行ユニットを備えていない

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

早く計算する能力は、人類にとって極めて重要になってきている。天候や気象状態の予測、医療分野(薬剤設計や断層撮影技術など)、国家防衛、地質学的調査、財務モデリング、インターネット検索、ネットワーク通信、多様な分野での科学研究、そして新しいコンピュータ・ハードウェアのデザインにおいても大量の計算を迅速に処理する能力に依存するようになってきている。複雑なナノ・スケールのシステムのコンピュータ支援設計や、見たり、聞いたり、理解したりするための消費者製品の開発などの分野においても

(3)

JP 2012-530966 A 2012.12.6

更なる進歩のために、更に高度な計算能力を経済的に展開していく必要が生じるものと考えられる。

#### 【0002】

ゴードン・ムーアの将来予測、すなわち、1ドル当たりの計算性能は2年毎に2倍に向上していくであろうという予測、は30年にわたって有効であることが立証され、これはある種の形態で、今後も続いていくものと考えられる。しかし、このような迅速な指数的な進歩にもかかわらず、現実には、シリコンによって利用可能な本来の計算能力は、ソフトウェアを利用可能にする速さよりもはるかに速い能力を有するまで成長している。言い換えれば、コンピュータ・ハードウェアの理論上の計算能力が指数関数的に成長を遂げてきているとはいえ、ソフトウェアがハードウェアにアクセスするために必要となるインターフェースのために、計算を実行するためにソフトウェアがハードウェアを使用する能力が、ハードウェアの理論上の最大計算能力に近づく段階において、制限されてしまうということである。

10

#### 【0003】

約10億個のトランジスタを内蔵し、クロック周波数がほぼ1GHzの最新のシリコン・マイクロプロセッサ・チップについて考えてみる。1回のサイクルで、そのチップはほぼ1つの有効な演算処理を稼働中のソフトウェアに配信する。例えば、一つの値がレジスタ間で移送され、別の値が加えられ、そして、乗算が完了する。今日ではクロック速度が1000倍程度早くなっているものの、このことは、マイクロプロセッサ・チップが30年前にやっていたことと大きく異なるものではない。

20

#### 【0004】

実際のコンピュータは、物理的なデバイスとして構築されており、コンピュータのマシンが構築される基礎となる物理学は、しばしば複雑で興味深い作用を提供するものである。例えば、シリコン金属酸化膜半導体電界効果トランジスタ(シリコンMOSFETトランジスタ)は、指数処理のような、興味深い非線形処理を行う機能を有するデバイスである。

2つのワイヤーの接点では電流が加算される。もし適切な形態に設定すれば、10億個のトランジスタとワイヤーは、基本的なコンポーネントの2、3の伝播遅延の間(設計全体が従来から行われているデジタル設計であるならば、「サイクル」と呼ばれる)に、計算に関係する10億の処理のかなりの割合を実行することができる。そして、今日のCPUチップは、ソフトウェアが、可能性のある10億の処理のかなりの割合を実行するのではなく、単に、サイクル毎に2、3の上述したような処理を実行できるようにするために、10億個のトランジスタを使用している。

30

#### 【発明の概要】

#### 【発明が解決しようとする課題】

#### 【0005】

本発明の実施例は、プロセッサあるいはその他のデバイスに関するものであり、プログラマブル及び/又は大規模並列処理プロセッサおよびその他のデバイスに関係する。

そして、このプロセッサあるいはその他のデバイスは、低精度ハイ・ダイナミック・レンジ(LPHDR演算)における数値を取り扱う演算処理(例えば、1又は2以上の加算、乗算、減算、除算を必ず含むものではないが、これらを含む可能性がある演算処理)を実行するように設計された処理要素を含む。

40

かかるプロセッサあるいはその他のデバイスは、例えば、単一のチップに実装されている。単一のチップに実装されているか否かにかかわらず、本発明のある実施例におけるプロセッサあるいはその他のデバイスで使用されるLPHDR演算要素の数は、従来の精度を有するハイ・ダイナミック・レンジの演算(32ビットまたは64ビット浮動小数点演算のような演算)を実行するために設計された当該プロセッサあるいはその他のデバイスで使用されている演算要素の数をはるかに超えるようになっている。

#### 【0006】

ある実施例においては、「低精度」演算要素は、少なくとも0.1%(1%の10分の1)の頻度で正しい結果とは異なる結果を生じさせるような演算処理を実行する。これは、広く採

50



(4)

JP 2012-530966 A 2012.12.6

用されているIEEE 754 単精度浮動小数点の標準規格に比べるとかなり悪い精度となっている。本発明におけるプログラマブル・プロセッサの実施例においては、このような異常な多数の演算エラーが生じて、適切に機能するアルゴリズムを使用してプログラミングが行われる。

また、ある実施例においては、処理要素はインプットを処理し、そして/又は、少なくとも100万分の1から100万までの広がりを持つレンジのアウトプットを出力する能力を有するという意味で、処理要素は「ハイ・ダイナミック・レンジ」を有している。

【図面の簡単な説明】

【0007】

図1は、本発明の一実施例であるSIMDプロセッサの全体設計を例示するものである。

10

【0008】

図2は、本発明の一実施例であるSIMDプロセッサの処理要素アレイを例示するものである。

【0009】

図3は、本発明の一実施例であって、プロセッサの処理要素アレイにおいて、処理要素がどのようにして当該プロセッサの他の部分とデータ通信するのかを例示するものである。

【0010】

図4は、本発明の一実施例である処理要素の設計を例示するものである。

【0011】

20

図5は、本発明の一実施例であるLPHDRデータのワード・フォーマットを例示するものである。

【0012】

図6は、本発明の一実施例であって、LPHDR演算ユニットの設計を例示するものである。

【0013】

図7は、オリジナルの画像を示すものである。

【0014】

図8は、本発明の一実施例であって、ブラー・カーネルによってぼやけさせた画像を示すものである。

【0015】

30

図9は、本発明の一実施例である浮動小数点演算を使用したRichardson Lucyのデコンボリューションによって生成された画像を示すものである。

【0016】

図10は、本発明の一実施例であって、付加ノイズ(fp+ノイズ)を有し、LPHDR浮動小数点演算を使用したRichardson Lucyのデコンボリューションによって生成された画像を示すものである。

【0017】

図11は、本発明の一実施例であって、LPHDR対数演算(lns)を使用したRichardson Lucyのデコンボリューションによって生成された画像を示すものである。

【発明を実施するための最良の形態】

40

【0018】

上述したように、今日のCPUチップは、内蔵するトランジスターを効率よく使っていない。例えば、10億個のトランジスターを内蔵する従来のCPUチップでは、各クロックサイクル毎に、ソフトウェアがわずかに数回の処理を実行できるようになっているに過ぎない。このことは極めて非効率なことであるが、当該技術の分野の通常の知識を有する者は、有効な理由であるとして幅広く受け入れられてきた理由により、このような手法でCPUを設計している。例えば、ここで述べた設計手法は、初期設計におけるソフトウェアの適合性にかかる要求事項(これは、しばしば本質的な要求事項である)を満たすものであった。さらに、この設計手法では極めて高い精度を実現するものであり、典型的には32ビットあるいは64ビット長の整数を使って正確な演算を行い、そして、32ビットと64ビット

50

(5)

JP 2012-530966 A 2012.12.6

の浮動小数点数を使ったかなり正確で、広く標準化された演算を行うようになっている。

多くのアプリケーションでは、この種の精度は必須のものである。その結果、従来のCPUでは、演算処理を実行するために100万個のオーダーのトランジスターを使って、典型的には上述した精度を提供できるように設計されている。

【0019】

しかしながら、精度に関しては感度が高くなく、トランジスター1個当たりのアプリケーションパフォーマンスという形態で、極めて大きな利得を得ることができる、経済的に重要なアプリケーションは多くある。そして、100万個のトランジスターが本来備えている計算能力のかかなりの割合を引き出す能力に基づき、この極めて大きな利得を得るのである。汎用コンピュータで用いられている現状のアーキテクチャーは、この能力を実現させることができないのである。

10

【0020】

典型的なマイクロプロセッサのような従来のコンピュータの弱点に関する理由から、より高度な性能を実現するために、別の種類のコンピュータが開発されてきている。このようなマシンには、単一命令ストリーム/複数データストリーム(SIMD: Single Instruction Stream/Multiple Data Stream)の設計手法、複数命令ストリーム/複数データストリーム(MIMD: Multiple Instruction Stream/Multiple Data Stream)の設計手法、フィールド・プログラマブル・ゲート・アレイ(FPGAs: Field Programmable Gate

Arrays)のような再構成可能なアーキテクチャー、そして、汎用コンピュータに応用された場合には、単一命令ストリーム/複数スレッド(SIMT: Single Instruction Stream/Multiple Thread)の設計手法として見ることもできるグラフィック・プロセッシング・ユニット(GPUs: Graphics Processing Units)の設計手法が含まれている。

20

【0021】

SIMDマシンは、データの集合体の処理を行う命令を備えたシーケンシャル・プログラムによって動作するようになっている。SIMDマシンには二つの主要な形式のものがおり、それはベクター・プロセッサとアレイ・プロセッサである。ベクター・プロセッサは、処理要素(あるいは処理要素の小さな集合体)を通じてデータを流す。データ・ストリームの各コンポーネントは同様に処理される。ベクター・マシンでは、多くの命令フェッチ/デコード処理(メモリから命令を持ってくる/命令を解釈すること)を省略すること、および処理のクロック・スピードが上がるようにプロセッサに連続送信する(パイプライン方式)ことにより、高速化を達成する。

30

【0022】

アレイ・プロセッサは、処理要素(PEs)のグリッドにデータを分配する。各処理要素は、独自のメモリを備えている。命令は中央制御装置から処理要素に、シーケンシャルに送信される。各処理要素(PE)は、各処理要素が持つローカル・データに対して送信された命令を実行する(しばしば、特定のサイクルでは、「アイドルング」というオプションを取り得る)。アレイ・プロセッサでは、シリコンを効率的に使用することにより高速化を達成する。すなわち、アレイ・プロセッサでは、多くの小さくてシンプルな実行ユニットを平行に動作させるために、1つだけの命令フェッチ/デコード・ユニットを使用している。

40

【0023】

アレイ・プロセッサは、幅広いビット幅、たとえば1、4、8ビット幅あるいはそれ以上のビット幅を有する固定小数点演算と、浮動小数点演算を使用して構築されてきた。ビット幅が小さければ処理要素を小さくすることができ、ほとんどの処理要素をコンピュータ内に収納できるようになるが、従来形式の演算を行うためには、多くの処理を順番に実行する必要が生じる。ビット幅が広くなれば、従来形式の演算処理を単一のサイクルで完結できる。実際のところ、ビット幅を広げることが望ましい。1ビット幅の処理要素を使ったコネクション・マシン-1やGoodyear超並列プロセッサのように、もともと小さなビット幅で設計されているマシンでは、高速演算をうまくサポートするために、より広いデータ・パスを志向して開発が行われ、32ビット浮動小数点演算用ハードウェアを

50



(6)

JP 2012-530966 A 2012.12.6

備えたコネクション・マシン-2や、Goodyearマシンの後を継いだ計算機であって、MasPar-1では4ビットの処理要素を用い、MasPar-2では32ビットの処理要素を用いた超並列計算機MasParのようなマシンが作り出された。

#### 【0024】

また、アレイ・プロセッサは、数値のアナログ表現を行い、計算を行う際にアナログ回路を使用するために設計されたものである。SCAMPはこのようなマシンに該当する。

これらのマシンは、低精度の演算を行うものであり、このマシンで行われる各処理の結果には、数パーセント程度のエラーが生じる。また、これらのマシンは、計算過程においてノイズを生じさせるので、計算を繰り返すことはできない。更に、これらのマシンでは、狭い範囲の値のみ扱うものであり、例えば、32または64ビット浮動小数点演算のハイ・ダイナミック・レンジを提供するというよりは、むしろ8ビット固定小数点数値に対応するものである。このような制約から、SCAMPは汎用コンピュータを指向したのではなく、画像処理および初期の生物学的視覚プロセスにおけるモデリング用として設計され、使用されているものである。このようなアプリケーション分野では、ハードウェアにおける演算処理のフルレンジ(全項目)を必要とするものではなく、例えば、SCAMPは、その設計の中から一般的な除算と乗算を省いている。

#### 【0025】

1980年代においては、SIMDマシンはポピュラーなものであったが、マイクロプロセッサの価格/性能が向上したので、設計者は、通信マイクロプロセッサの大きな集合体からなるマシンを構築し始めた。MIMDマシンは、高速で、そのコンポーネントであるマイクロプロセッサに匹敵する価格/性能を有するが、MIMDマシンは、そこで使用されているソフトウェアに対して相対的に低いトランジスタ1個当たりの計算性能しか実現できないという点において、そのコンポーネントと同程度の非効率さを有している。

#### 【0026】

フィールド・プログラマブル・ゲート・アレイ(FPGAs)は、汎用デジタル要素の大きなグリッドを備えた集積回路であり、デジタル要素間のワイヤリングは再構築可能となっている。このデジタル要素は、もとは「AND」と「OR」ゲートのようなシングル・デジタル・ゲートであるが、ブール関数によって、例えば、6入力1出力のマップ(map)にプログラミングできる大きな要素として開発された。このアーキテクチャーは、FPGAが外部ソースから幅広い種類のデジタル計算を実行できるように構成されている。そしてこのデジタル計算は、デバイスがCPUの計算を促進するためのコプロセッサ(副処理装置)として使用できるようにするものである。

しかし、整数、と特に浮動小数点数に対する乗算や除算のような演算処理には、多くのゲートが必要になり、FPGAの汎用のリソースの大部分を使い切ってしまうことになる。このような理由から、最新のFPGAsでは、FPGAsのかかなりの比率の領域を、数ダースあるいは数百の乗算ブロックを提供するために割り当てている。そしてこの数ダースあるいは数百の乗算ブロックは、乗算を必要とする計算のために、汎用のリソースの代わりに使用できるようになっている。これらの乗算ブロックは、典型的には18ビットあるいはそれ以上の幅の整数の乗算を実行するものであって、多くのトランジスタを使用するものである。そして、これらの乗算ブロックは、汎用CPUの一部として機能する乗算回路と同様である。

#### 【0027】

既存のフィールド・プログラマブル・アナログ・アレイ(FPAAs)は、FPGAsに類似しているが、FPAAsの設定可能な要素はアナログ処理を行うようになっている。一般に、このようなデバイスは、神経回路のモデル化を助けるような、信号処理を行うためのものである。FPAAsは比較的低精度であって、比較的低いダイナミック・レンジを有しており、計算過程でノイズを生じるものである。FPAAsは、汎用コンピュータとして、あるいは汎用コンピュータに使用されることを意図して設計されたものではない。例えば、当該技術の分野の通常の知識を有する者には、FPAAsは、ハイ・パフォーマンスのデジタル・コンピュータ上で使用されている、浮動小数点演算を使用した複雑な多種のアルゴリズム

(7)

JP 2012-530966 A 2012.12.6

を実行することができるマシンとしては見られてこなかった。

【0028】

最後に、グラフィックス・プロセッシング・ユニット(GPUs)には、パーソナル・コンピュータ用として高速のグラフィックス能力を提供するために開発されたさまざまな並列プロセッサがある。GPUsは、特定のタスクにおいて極めてハイ・パフォーマンスな性能を発揮する標準的な浮動小数点計算能力を提供している。GPUsの計算モデルは、往々にして、数千からなるほぼ同一の計算スレッドに基づいている(SIMT)。その点について、それはSIMDに似た内部計算エンジンの集合体によって実行され、各エンジンは、遅い外部DRAMメモリがデータを提供する作業を実行するために、結果の出力先を決め、あるいは出力先を変更する。標準的な浮動小数点演算を実行するその他のマシンのように、GPUsは、その演算のために多くのトランジスターを使用している。GPUsは、上述したような意味において、汎用のCPUsと同様に、これらのトランジスターを無駄に使っている。

【0029】

あるGPUsは16ビット浮動小数点数値をサポートしている(往々にして、「ハーフ」フォーマットと呼ばれる)。NVIDIAやAMD/ATIのような現在のGPU製造業者は、通常の32ビットのRGBAフォーマットよりも高度なダイナミック・レンジを備えたレンダリング画像にとって、この能力は有用であると述べている。そして、32ビットのRGBAフォーマットは色毎に8ビットの固定小数点データを使用し、色のコンポーネントごとに32ビットの浮動小数点を使用することに対して、使用領域を節約している。特殊効果映画ファームであるインダストリアル・ライト・アンド・マジック社(ILM)は、彼らのOpenEXR標準において独自に同一の画像を定義した。そして、ILM社は、ハイ・ダイナミック・レンジ(HDR)画像ファイル・フォーマットは、コンピュータ画像アプリケーションの分野で使用するために、インダストリアル・ライト・アンド・マジック社(ILM)によって開発されたものであると述べている。Wikipedia(2008年後半)は、16ビット浮動小数点画像について以下のよう

に述べている。すなわち、  
「このフォーマットは、OpenEXR、OpenGL、B3DXを含むいくつかのコンピュータ・グラフィックス環境において使用されている。8ビットまたは16ビット2進整数に対する利点は、拡大されたダイナミック・レンジによってハイライト部分と影の部分においてより詳細に保存できるということである。32ビット単精度2進数フォーマットに対する利点は、記憶容量およびバンド幅が半分になることである。」

【0030】

グラフィックス・プロセッサが16ビット浮動小数点をサポートしているとき、そのサポートは32ビット浮動小数点を一緒にサポートするものであり、さらには、64ビット浮動小数点をも一緒にサポートするものである。すなわち、16ビット浮動小数点フォーマットは、それを必要とするアプリケーションでサポートされ、従来のグラフィックス・アプリケーションやいわゆる汎用GPUアプリケーションにとっても必要であると考えられているため、さらに高精度のフォーマットもサポートされている。従って、既存のGPUsでは、実質的なリソースを32ビット演算(さらには64ビット演算)に充てており、上述したような意味においてトランジスターの浪費を行っている。

【0031】

上述した種々のアーキテクチャーの全ては、従来のプロセッサの設計で利用できる性能よりも更に高度な性能をシリコンから引き出すために行われてきた試みである。しかし、従来のプロセッサの設計者は、マシンの性能を向上させようとする莫大な数のトランジスターを使用する必要がある、苦しんできた。このようなマシンは、歴史的および経済的に、インテルx86命令セットのような既存の大きな命令セットをサポートするために、往々にして必要になる場合がある。トランジスターの数を2倍にしても性能を2倍にすることはできないという、収穫逨減の法則に基づく理由から、これは困難な問題である。このような設計者の苦悩の一面は、演算処理の精度を向上させることであった。これは、トランジスターは十分であり、プロセッサがもともと長い数値(例えば64ビットの数値)をサポートしていたのであれば、あるアプリケーションでは顕著にスピー

(8)

JP 2012-530966 A 2012.12.6

ドアップを図れたからである。8ビットから16ビット、32ビット、64ビットまでの固定小数点数の精度の向上によって、及び32ビットから64ビット、あるときには128ビットまでの浮動小数点数の精度の向上によって、プログラマーは高精度という点について考えるようになり、コンピュータ・プロセッサというものは、かかる精度を提供するものだという前提に立って、アルゴリズムの開発を行うようになってきている。なぜなら、精度はシリコン・チップの新しい世代の全てであり、何ら拘束されるものではないからである。

#### 【0032】

本発明の実施例は、上述した方法とは根本的に異なったアプローチによって効率的に計算能力を提供するものである。特に、本発明の実施例は、コンピュータ・プロセッサあるいはその他のデバイスに係るものであって、それは、計算(例えば、演算処理)を実行するために低精度ハイ・ダイナミック・レンジ(LPHDR)の処理要素を使用するものである。

#### 【0033】

LPHDR演算の一種では、約0.1%の精度であって、100万分の1から100万までの値を表す。もし、これらの値が表わされ、浮動小数点演算の方法により取り扱われるならば、これらの数は、10ビット+符号ビットを超えない2進法の仮数と、少なくとも5ビット+符号ビットの2進法の指数を有するようになる。しかし、このような浮動小数点数の乗算や除算を行う回路は比較的大きなものになってしまう。別の実施例の一つの例は、浮動小数点数の対数表示を使用するものである。このようなアプローチでは、浮動小数点数を表示するために必要となるビット数は同じであるが、乗算や除算は、対数表示の加算、および減算としてそれぞれ実行される。加算や減算は、以下に説明するように効率的に実行することができる。その結果、演算回路の領域を比較的小さくすることができ、そして多くの計算要素をシリコン上の与えられた領域の中に配置することが可能となる。このことは、マシンが単位時間当たり、あるいは単位電力当たり、より多くの処理を実行することができることを意味する。そしてこのことは、マシンがLPHDRのフレームワーク内で表現できる計算の利点を享受できるということを意味する。

#### 【0034】

別の実施例は、アナログ表現とプロセッシング・メカニズムを利用するものである。LPHDR演算のアナログ演算の実行は、デジタル演算の実行よりも優れたものになるという可能性を有している。なぜならば、アナログ演算は、デバイスの挙動のデジタル的挙動を示す部分のみを利用するのではなく、トランジスタあるいはその他の物理的なデバイスが本来備えているアナログ的物理挙動を利用しようとするものだからである。このように、デバイスが本来備えている能力の全体を利用することは、LPHDR演算を実行するために必要なメカニズムを小規模なものにすることができる。近年、シリコン回路の技術分野では、アナログ的な手法は、デジタル手法によって置き換えられてきている。これは、アナログ設計に比べ、デジタル設計の方が容易であるという理由によるものである。

また、アナログ技術に比べデジタル技術は継続的に急激な拡張を遂げてきた(ムーアの法則)という理由もある。特に、ディープ・サブミクロンの領域では、大きな寸法を有する前世代のトランジスタのように、アナログ・トランジスタはもはや機能しない。

良く知られている挙動の近年におけるこのような変化は、アナログ設計をさらに難しくしている。

しかし、デジタルトランジスタは、実際のところ、デジタル的手法で使用されているアナログ・トランジスタであり、デジタル回路は、実際のところ、完全に「ON」の状態と、完全に「OFF」の状態に、トランジスタをスイッチングするように設計されたものである。デジタル技術の拡張は継続するが、このようにトランジスタを使用することは、アナログ挙動の現実と直面し始めることになる。デジタル的使用方法におけるトランジスタの拡張は、行き詰るか、あるいは、デジタル設計者はアナログの問題を認識し、アナログの問題を取り扱うことが必要になると考えられている。このような理由のために、LPHDR演算におけるデジタル的な実施例はもはや、容易であって、信頼でき、かつ

(9)

JP 2012-530966 A 2012.12.6

拡張可能なものとは言えない。そして、LPHDR演算におけるアナログ的な実施例は、商業上のアーキテクチャーとして優位に立つようになる可能性がある。

【0035】

LPHDR処理要素は、比較的小さいので、単一のプロセッサあるいはその他のデバイスに、非常に多数のLPHDR処理要素を収納することができ、かつ互いに並行処理でき、そしてその結果、大規模並列処理LPHDRプロセッサまたはその他のデバイスを構成することができる。このようなプロセッサやその他のデバイスが、当該技術分野における通常の技術常識を有するものによって、汎用的な計算を行う手段として説明されたり、実際に使用されたりしたことはなかった。これには少なくとも2つの理由がある。一つは、当業者の間では、以下のことが広く考えられていた。すなわち、大規模並列処理が行われるか否かにかかわらず、LPHDR計算、特に大量のLPHDR計算は、中程度の汎用計算用の回路基板として実用的でないということである。二つ目には、大量、高精度計算はコンパクトな計算処理ユニットによって処理することができるので、マシン内にある処理要素間、マシンへの入力、そしてマシンの外部のバンド幅を増加させることなく、単一チップあるいは単一マシン上で大量、高精度計算を行うことは有用ではない、と当業者の間で広く考えられていた。なぜならば、計算はワイヤリングによって制限を受け、計算はコストをかけずに実行することができると考えることができるからである。

【0036】

単一チップあるいは大規模並列処理マシン上で大量の計算を行うことは有用ではなく、大量のLPHDR演算は却って悪い結果を生むものである、という考え方にもかかわらず、ここで開示する本発明の実施例では、大規模並列LPHDR設計は実際には有用であり、少なくとも数種類のアプリケーションにおいては顕著な実利的な利点を提供できることを実証している。

【0037】

結論として、最新のデジタル演算システムは、高精度演算を提供することができるが、この高精度のためにコスト高になる。低精度の乗算を実行するためには、わずかな数のトランジスタしか必要としないにもかかわらず、最新の倍精度浮動小数点乗算器は、100万オーダーのトランジスタを必要とする。最新のアプリケーションは高精度処理を必要としている、ということが当業者の間では広く信じられているにもかかわらず、実際には、ある種の有用なアルゴリズムはもっと低精度であっても適切に機能する。その結果、このようなアルゴリズムは、本発明の実施例を適用したプロセッサまたはその他のデバイスによって実行することができる。そして、このことは乗算のための僅かなトランジスタと、加算のためのワイヤ接続を使用するという最終目標の実現に近づくものである。更に、このことは、比較的少量の物理的リソース(単一のシリコン・チップのようなもの)によって、大規模並列演算を可能とするものである。ある専門分野に特化したタスクは、低精度でも機能するが、汎用コンピュータにおいて今日典型的に実行されているような比較的汎用的な計算は、低精度でも実施できるものであるということは、自明なことではなく、実際には当業者からは明らかに誤った考え方であると見られてきた。

しかし、実際のところ、ある種の有用で、重要なアルゴリズムは、大規模並列計算のフレームワークにおいて、32ビット精度以下であっても適切に機能させることができる。そして、本発明のある実施例では、そのようなアルゴリズムをサポートしており、したがって、トランジスタを効率的に使用するようになっている。その結果、従来のコンピュータと比較して、スピード、消費電力、及び/又はコストを向上させている。

【0038】

以下に、本発明の実施例を適用した種々の計算デバイスについて説明する。ここで説明する実施例のいくつかは、SIMDコンピュータ・アーキテクチャーを例示するものである。MIMDアーキテクチャ、プログラマブル・アレイ・アーキテクチャー(FPGAsやFPAAsなど)、あるいはGPU/SIMTアーキテクチャなどその他のアーキテクチャを使用することもできる。ここで開示する技術は、例えば、前述したような既存のアーキテクチャーを備え



(10)

JP 2012-530966 A 2012.12.6

たプロセッサ又はその他のデバイスを使用し、ここで開示するいずれかの方法で使用されているLPHDR演算ユニットで、プロセッサ又はその他のデバイスに収納されている特定のあるいは全ての既存の演算ユニットを置き換えるか、増補することによって、実行することができる。

しかし、本発明の実施例に従って実施されるデバイスは、既存のプロセッサの設計をそのまま使用してスタートする必要はかならずしもなく、まさにここで述べたアーキテクチャあるいはその他のアーキテクチャの中にLPHDR演算ユニットを含めるために、本発明の実施例に従って実施されるデバイスを最初から設計するようにしても良い。

【0039】

例えば、本発明の実施例は、特殊なSIMDコンピュータである、アレイ・プロセッサのアーキテクチャを使用して実施することもできる。科学文献や商業文献に開示されているアレイ・プロセッサには多くの種類のものがあり、かつ特殊なものもある。例示するならば、イリアック4(Illiack 4)、Connection Machine 1 and 2、Goodyear MPP、MasPar等のコンピュータがある。

【0040】

しかし、本発明の実施例は、SIMDコンピュータとして実施する必要は必ずしもない。

例えば、本発明の実施例は、FPGAs、FPAAs、あるいは一群の処理要素に柔軟な接続性を提供する関連アーキテクチャとして実施するようにしても良い。例えば、本発明の実施例は、GPU/SIMTs、MIMDs、その他のものとして実施するようにしても良い。例えば、本発明の実施例は、従来のアーキテクチャに比べ、少ないリソース(トランジスタの数や体積)を使って計算するために、LPHDR演算処理要素を使用したどんな種類のマシンとしても実施することができる。更に、本発明の実施例において言及する「処理要素」は、LPHDR処理を行うものであるか否かは別として、いずれかの種類の演算実行部としてより一般的に解釈すべきである。

【0041】

SIMDコンピューティング・システム100の例を図1に示す。このコンピューティング・システムは、多数の処理要素(PEs)の集合体 104を含んでいる。このコンピューティング・システムは、往々にして、コントロール・ユニット(CU) 106、I/Oユニット(IU) 108、種々の周辺デバイス 110、そしてホスト・コンピュータ 102を備えている。

処理要素(PEs)の集合体 104は、2次元的な配列、アレイ、グリッド、その他の特定のレイアウトをとる必要は必ずしもないが、ここでは「処理要素アレイ」(PEA) 104と呼ぶ。あるマシンでは、Goodyear MPPの「ステージング・メモリ」と呼ばれる追加的メモリ・システムのような、追加的なコンポーネントを備えている。しかし、このような追加的コンポーネントは、コンピュータにとって本質的なものではなく、本発明の実施例を理解するために必要なものでもない。したがって、説明を分かりやすくするために、ここでは、このような追加的コンポーネントは、省略することにする。

本発明の一つの実施例は、図1に示すようなSIMDコンピューティング・システムであり、本明細書における用語を使えば、処理要素アレイ(PEA) 104中における一つ又はそれ以上(例えば、全て)のPEsはLPHDR処理要素になる。

【0042】

ホスト・コンピュータ102はコンピューティング・システム100の全体制御を行う責任を負っている。ホスト・コンピュータ102は従来タイプのユニ・プロセッサであり、シリアル(あるいはほとんどシリアル)な演算を行うものである。もちろん、ホスト・コンピュータ102は、いろんな種類の並列処理を含む更に複雑な構造を備えるようにすることもできる。実際、単一のマシンにおいて、複数の計算アーキテクチャを組み合わせた異機種環境の計算システムを採用することは、本発明の実施例にとっては好ましい使用方法である。

【0043】

ホスト・コンピュータ102の目的は、処理要素アレイ(PEA) 104に、有用な方法で、大量の計算を実行させることである。ホスト・コンピュータ102は、PEsに、典型的には各

PEに局所的に記憶されたデータに対する計算を、互いに並列的に実行させることにより、大量の計算を実行させる。例えば、多数のPEsが存在するときには、各単位時間当たりに、大量の計算が実行される。

#### 【0044】

処理要素アレイ(PEA) 104中のPEsは、ホスト・コンピュータ102が実行する計算と同様な速度で、個々の計算をラフに行うことができるようになっている。このことは、ホスト・コンピュータ102又は処理要素アレイ(PEA)104の最小タイム・ステップと同じ細かいタイム・スケールで、ホスト・コンピュータ102が処理要素アレイ(PEA) 104を制御するように試みることは非効率であることを意味するものである。(この最小タイム・ステップは、従来のデジタル設計におけるクロック周期に相当する。) このような理由のために、特化したコントロール・ユニット(CU)106をアーキテクチャの中に含めるようにしても良い。コントロール・ユニット(CU) 106の主要タスクは、命令メモリからの命令を回収し復号することであり、このことは、概念的にはコントロール・ユニット(CU) 106の役割であり、部分的に復号した命令を処理要素アレイ(PEA) 104の全てのPEsに発信する。(このことは、コントロール・ユニット(CU) 106のソフトウェアによって、全てのPEsに対してほぼ同時に生じる出来事として観察するようにしても良い。そして、それは文字通りに同期したものである必要はない。実際のところ、複数の命令を異なった実行段階に、処理要素アレイ(PEA) 104全体に亘って波面が伝播するように、同時に伝播させるような、同期させた設計にすることは効率的である。)

#### 【0045】

コントロール・ユニット(CU) 106を含む設計では、ホスト・コンピュータ102は、典型的には、処理要素アレイ(PEA) 104に対する命令(プログラム)をコントロール・ユニット命令メモリ(図1には図示せず)にロードする。そして、コントロール・ユニット(CU) 106に対して、プログラムを解釈し、処理要素アレイ(PEA) 104に命令に従った計算を実行させる。例えば、このプログラムは、PEs内およびPEs間におけるデータの移動、論理的処理、演算処理等、およびコントロール・ユニット(CU) 106内での制御の流れ処理と一緒に類似の処理を行うための、その他の命令を備えた、一般的な機械言語プログラムに類似したものであっても良い。従って、コントロール・ユニット(CU) 106は典型的なタイプのプログラムを実行することができるが、処理要素アレイ(PEA) 104に対して大規模な並列処理命令を発生させる能力をも有している。

#### 【0046】

コントロール・ユニット(CU) 106および処理要素アレイ(PEA) 104の中へデータを送り、そしてこれらからデータを取り出すために、I/Oユニット(10U) 108がコントロール・ユニット(CU) 106および処理要素アレイ(PEA) 104とホスト・コンピュータ102、ホスト・コンピュータ・メモリ(図1には図示せず)、外部記憶装置(例えばディスク・ドライブ)や計算結果を可視化するためのディスプレイ装置や特別な広いバンド幅を有する入力装置(例えば、ビジョン・センサー)のようなコンピュータ・システムの周辺デバイス 110の間のインターフェースをとるようにしても良い。ホスト・コンピュータ 102よりもはるかに速くデータを処理できる処理要素アレイ(PEA) 104の能力は、データ移送のある部分をホスト・コンピュータ 102をバイパスして実行できるようにするために、I/Oユニット(10U) 108にとって有用になる。更にホスト・コンピュータ 102は周辺デバイス 110と通信するための独自の経路を有するようにしても良い。

#### 【0047】

図1に示す特定の実施例は、単に例示のために示したものであり、本発明を限定するためのものではない。例えば、代替的なものとして、コントロール・ユニット(CU) 106によって実行される機能は、コントロール・ユニット(CU) 106を省略してホスト・コンピュータ 102によって代行させることもできる。コントロール・ユニット(CU) 106は、処理要素アレイ(PEA) 104から離れたハードウェア(例えば、チップから外れたハードウェア)として設置することもできるし、あるいはコントロール・ユニット(CU) 106は、処理要素アレイ(PEA) 104の近く(例えば、同一チップ内)に配置することもできる。

(12)

JP 2012-530966 A 2012.12.6

I/Oは、I/Oユニット(I/O) 108を省略し、コントロール・ユニット(CU) 106を経由した経路とすることもできるし、図示するように、別途設けたI/Oユニット(I/O) 108を経由するようにすることもできる。更に、ホスト・コンピュータ 102はオプションであって、例えば、コントロール・ユニット(CU) 106がCPUを備えるようにしても良いし、あるいは、ホスト・コンピュータ 102によって実行できる機能を代替できる機能を有するコンポーネントを備えるようにしても良い。図1に示す周辺デバイス 110はオプションである。図1に示す設計では、中間レベルの局所記憶装置を提供するGoodyear MPPの「ステージング・メモリ」のような特別のメモリを備えるようにすることができる。

例えば、このようなメモリは、処理要素アレイ(PEA) 104の処理要素(PEs)からメモリに比較的高速に並列アクセスするために、3D組立技術を使用したLPHDRチップに接続することができる。

10

## 【0048】

処理要素アレイ(PEA) 104は、コントロール・ユニット(CU) 106やI/Oユニット(I/O) 108、あるいはその他の可能なメカニズムとの間で通信できるのに加え、処理要素アレイ(PEA) 104は、それ自身アレイ内でデータを移動させる手段を備えている。例えば、処理要素アレイ(PEA) 104は、処理要素(PEs)から最も近くに隣り合う処理要素(PEs)に対してのみデータを移動させることができるように駆動することができる。すなわち、データ移送のための距離は、もはや存在しない。図2および3は、本発明の実施例を示すものであって、上述したようなアプローチを採用した実施例を示す。ここでは、最も近くに隣り合う処理要素(PEs)は、4つの隣接した処理要素(PEs)であり、北、東、西、南の各処理要素(PEs)からなり、NEWS設計と呼んでいる。例えば、図2は、処理要素アレイ(PEA) 104における処理要素(PEs)の小集合、すなわちPE 202、PE 204、PE 206、PE 208、およびPE 210を示したものである。コントロール・ユニット(CU) 106がデータ移動命令を発すると、全ての処理要素(PEs)は、個々の処理要素(PEs)の最も近くに隣り合う特定の処理要素(PEs)からのデータにアクセスするか、あるいは個々の処理要素(PEs)の最も近くに隣り合う特定の処理要素(PEs)へ向けてデータを送るようになっている。

20

例えば、すべての処理要素(PE)が、西側に隣り合う処理要素(PE)内の特定のデータ数値にアクセスし、それを自らの局所記憶装置の中にコピーする。アナログ処理の実施例のような、ある実施例では、このような種類のデータ移送は、コピーされた数値の劣化が生じることになる。

30

## 【0049】

図3は、I/Oユニット(I/O) 108とのデータ接続を備えた処理要素(PE) 302を示したものである。処理要素(PE) 302は、北側で処理要素(PE) 304、東側で処理要素(PE) 306、南側で処理要素(PE) 308、西側で処理要素(PE) 310に接続されている。しかし、処理要素アレイ(PEA) 104の内部からI/Oユニット(I/O) 108へ出ていく駆動信号は、通常、物理的に比較的大きな駆動回路あるいはアナログ・メカニズムを必要とする。全ての処理要素(PE)において、このようなものを備えることは、ハードウェアの実行技術上の利用できるリソースの多くの部分(VLSI領域のようなもの)を使い尽くすことになってしまう。更に、全ての処理要素(PE)からI/Oユニット(I/O) 108への独立した接続を備えるようにすることは、多くの接続と長い接続を備えることを意味し、これもまた利用できるリソースの多くの部分を使い尽くすことになってしまう。このような理由のため、処理要素(PEs)とI/Oユニット(I/O) 108の間の接続は、処理要素アレイ(PEA) 104の端にある処理要素(PEs)に限定するようにしても良い。この場合、処理要素アレイ(PEA) 104からのデータを取得し、そして処理要素アレイ(PEA) 104へデータを送り出すために、データは、処理要素アレイ(PEA) 104の端にある処理要素(PEs)において読み書きされ、そして処理要素アレイ(PEA) 104の端にある処理要素(PEs)と処理要素アレイ(PEA)

40

104の内部側にある処理要素(PEs)との間でデータを移動させるために、コントロール・ユニット(CU) 106の命令が実行される。このような設計では、直接的な接続を利用して、I/Oユニット(I/O) 108から処理要素アレイ(PEA) 104の内側にあるいずれかの処理要素(PE)に向かってデータが送られていくようになっているが、I/Oユニット(I/O) 10

50



(13)

JP 2012-530966 A 2012.12.6

8によって読み込むことができる、処理要素アレイ(PEA) 104の端にある処理要素(PEs)にデータをおくするためには、コントロール・ユニット(CU) 106を利用してデータを読み込むようにしても良い。

【0050】

コントロール・ユニット(CU) 106と処理要素アレイ(PEA) 104の間の接続には、類似のバリエーションがある。ある設計では、命令をほぼ同時に全ての処理要素(PEs)に伝える能力を備えるようにしても良いが、別のアプローチでは、命令が徐々に伝播(例えば、離散的なタイム・ステップに基づくずらしによる)して処理要素アレイ(PEA) 104を横断し、処理要素(PEs)に命令が届くようにしている。本発明の実施例として実行される、ある種のSIMD設計においては、処理要素アレイ(PEA) 104の各処理要素(PE)のOR結線状態またはAND結線状態がコントロール・ユニット(CU) 106によって、ほぼ1命令時間の遅れで、読み取ることができるようになっている機器を備えている。

【0051】

このことについて良く研究された種々の論文が多数存在し、それらのうちのいくつかについては、本発明の実施例の中に取り込まれている。例えば、8WAYのローカル・インターコネクトのようなインターコネクトを使用することもできる。ローカル・インターコネクトでは、ディスタンス1であるもののほか、ディスタンス4や16となるもののように、ディスタンスのホップ数が種々の値をとるものが混在している。外側の端部では、トラス(torus)やねじれトラス(twisted torus)のようなトポロジーを使用して接続することができる。ローカル・インターコネクトに代えて、あるいはローカル・インターコネクトに加えて、ハイパーキューブ・デザインのような、更に複雑なグローバル・インターコネクトを使用することもできる。更に、更に大きな処理要素アレイ(PEA) 104を作りだすために、処理要素アレイ(PEA) 104 (例えば、チップ)は、複製する(例えば、回路のボード上に敷設する)ことによって物理的に実装することができる。PEAsのコンポーネントは必ずしもグリッド状の配列になっている必要はないが、複製することによってシンプルなグリッド状の配列あるいはその他の配列を形成することができる。

【0052】

図4は、処理要素(PE) 400(これは処理要素アレイ(PEA) 104中の処理要素(PEs)の一つまたは複数の機能を果たすものである)の一設計例を示したものである。処理要素(PE)

400はローカル・データを記憶する。このローカル・データのための記憶容量は設計毎に顕著に変化するものである。それは、処理要素(PE) 400を製造するために利用することができる実施技術に大きく依存している。めったに変更されない数値(コンスタント:Constants)は、頻繁に変更される数値(レジスター)よりも使用場所が少なく済む。そして、ある種の設計では、レジスターよりも多くのコンスタントを提供するようにしている。例えば、コンスタントに対しては単一のトランジスタ・セル群(例えば、フローティング・ゲート・フラッシュ・メモリー・セル)を使い、レジスターに対しては、複数のトランジスタ・セル群(例えば、6-トランジスタSRAMセル)を使ったデジタル回路の実施例がある。アナログ回路の実施例では、この状況は往々にして逆転する。この場合には、コンスタントの長期保存の安定性を確保するために、キャパシタンスに対する十分な領域が必要となる。そして、係る実施例では、コンスタントよりもレジスターを多く備えるようになっている。典型的には、各処理要素(PE)におけるレジスターおよびコンスタントに記憶される記憶容量は数10あるいは数100の演算値になるが、これらの記憶容量は設計者によって調整できるものである。例えば、ある設計では、レジスターの記憶装置は備えているが、コンスタントの記憶装置は備えていない。そして、ある設計では、各処理要素(PE)に記憶される数値は、数千あるいはそれ以上の多数の数値となる。このようなバリエーションの全ては、本発明の実施例に反映されている。

【0053】

各処理要素(PE)は、各処理要素(PE)のローカル・データを処理する必要がある。このため、処理要素(PE) 400の中には、データ・パス402a-i、ルーティング・メカニズム(マルチプレクサMUX 404のようなもの)、および論理処理と演算処理の結果を収集するための

(14)

JP 2012-530966 A 2012.12.6

コンポーネント(論理ユニット406及びLPHDR演算ユニット408のようなもの)が存在する。

LPHDR演算ユニット408は、本明細書で使用する用語である、LPHDR演算処理を行う。処理要素(PE) 400によって受け取り、処理要素(PE) 400によって出力され、処理要素(PE) 400によって処理された入力、出力、および処理途中の数値は、例えば、数値を表す電気信号の形態をとる。

#### 【0054】

また、処理要素(PE) 400は、図4に示すマスク410として示す、一つまたはそれ以上のフラッグ・ビットを備えるようにしてもよい。マスク410の目的は、いくつかの処理要素(PEs)を有効にし、コントロール・ユニット(CU) 106によって出された命令を無視するために、その処理要素(PEs)では特定のマスク・ビットがセットされる。このようにすることによって、処理要素アレイ(PEA) 104の中の全ての処理要素(PEs)の通常のロック・ステップ動作のバリエーションを作り出すことができるようになっている。例えば、コントロール・ユニット(CU) 106が、各処理要素(PE)が各処理要素(PE)のマスク410を、処理要素(PE)内の特定のレジスターが正であるか負であるかによってリセットしたりセットしたりする命令を出すことができる。例えば、演算命令のような後続の命令には、マスク410がリセットされた処理要素(PEs)によってのみ命令が実行されるべきであるとの意味を持たせたビットを備えるようにすることができる。その処理要素(PE)内の特定のレジスターが負であるかどうかによって、その組み合わせは、各処理要素(PE)において演算命令が条件付きで実行されるという効果を有するようになる。従来のコンピュータの比較命令と同じように、マスクをセットし、そしてクリアのためのメカニズムとしては、多くの設計上の選択肢がある。

#### 【0055】

処理要素(PEs)における処理は、コントロール・ユニット(CU) 106から受信した制御信号412a-dによって制御される。これら4つの制御信号は図4に示されているが、これらは単なる例示であって、これらに限定されることを意味するものではない。このメカニズムの詳細については示していないが、制御信号412a-dは、データ・バスに送るべき、処理要素(PE) 400内のレジスター・メモリの数値やコンスタント・メモリの数値、あるいは隣接する処理要素(PE)を特定する。そしてこのような処理は、論理ユニット406や演算ユニット408やその他の処理メカニズムによって実行され、その結果はレジスターに記憶される。そして制御信号412a-dは、セットやリセットの方法、およびマスク410を使用すること等を特定する。これらのことについては、SINDプロセッサの文献に詳細に記述されている。

#### 【0056】

処理要素(PE) 400と処理要素アレイ(PEA) 104の設計に関するバリエーションには多くの可能性が存在すると共に、これらは全て本発明の範囲に包含される。デジタルの処理要素(PEs)では、シフター、参照テーブル、その他上記文献にも記載されているような多くのその他のメカニズムを備えることができる。アナログの処理要素(PEs)では、時間ベースの演算子、フィルター、グローバル・ブロードキャスト・信号を備えたコンバータ、その他上記文献にも記載されているような多くのその他のメカニズムを備えることができる。処理要素アレイ(PEA) 104は、デジタルの処理要素(PEs)に対しては、OR結線またはAND結線のようなグローバル・メカニズムを備え、アナログの処理要素(PEs)に対しては、SUM結線を備えることができる。再度述べるが、デジタルおよびアナログのコンピュータ・アーキテクチャーに関する文献には多くのバリエーションが詳細に記載されている。

#### 【0057】

例えば、加算と乗算の他and/or以外のLPHDR処理をサポートすることもできる。例えば、乗算と関数(1-X)だけを実行することができるマシンは、概算加算およびその他の演算処理に使用することができる。LPHDR処理の集合体は、当該技術分野における通常の知識を有する者に周知の技術を使用して、加算、乗算、減算及び除算のようなLPHDR演算処理を概算するために使用される。

(15)

JP 2012-530966 A 2012.12.6

【0058】

本発明のユニークな実施例の一つの観点として、処理要素(PEs)の中にLPHDR演算メカニズムが含まれている。係るメカニズムの実施例について、以下に説明する。

【0059】

LPHDR演算ユニット408に関する一つのデジタルの実施例では、デジタル(2進数)表現された数値が処理される。あるデジタルの実施例では、これらの数値は対数表示される。

係る表現方式は対数システム(LNS)と呼ばれ、当業者には良く知られたシステムである。

【0060】

対数システム(LNS)では、数値は、符号と指数によって表現される。対数システム(LNS)では、対数に対する暗黙的な低が存在し、それはデジタルなハードウェアによって機能している場合には、典型的には2である。本実施例では、2の低が例示として使用されている。結果的に、数値Bは符号と低2の対数で表現され、数値bはその絶対値として表現される。最大でも1%の表示誤差を有する数値では、変化し得る小数部が数値Bにおける約1%の変化に相当する十分な精度で表示される。もし、分数が6ビットを使用して表現されるならば、その小数部を1だけ増減することは、数値Bに2の64乗根(ほぼ1.011の値をとる)を乗じたり、除したりすることに該当する。このことは、本実施例では、数値は約1%の乗法誤差を持って表示される可能性があるということを意味する。従って、この実施例では、小数部の表示に6ビットが割り当てられている。

【0061】

更に、本実施例において処理される値の領域は、ハイ・ダイナミック・レンジを有している。10億番目から更に10億までの絶対値を表すために、対数の整数部は10億の低を2とする正負の数値を表すことができる十分な長さでなければならない。その対数は、約29.9である。本実施例では、対数表示の整数部は、0から31の数値を表すために5ビットの長さになり、これで十分である。その他に、指数部の符号ビットが必要になる。負の対数は、2の補数表現を使用して表される。

【0062】

対数システム(LNS)では、0の値は負の無限大になる。この特別な値を正確に表現するために、表示方法を選択することができる。しかし、演算回路によって使用されるリソースを最小化するために(例えば、領域を最小化するために)、本実施例では、0を負側の最も大きい数値、-32によって表示しており、これは2の補数ビット表現、

「100000 000000」に対応し、約 $2.33E-10$ の値を意味するものである。

【0063】

計算を実行するとき、処理が合理的な値を導き出すことができない状況が生じる場合がある。例えば、非常に大きな二つの数値を掛け合わせたり、加えたりするときや、ゼロ(またはゼロに近い数値)で割り算を行うときのように、数値が大きすぎて選択したワード・フォーマットで表現することができないような場合である。このような問題を解決するための共通的なアプローチとして、得られた値に、「数値ではない:Not A Number(NAN)」というマークを付すようにし、問題が発生して処理途中において「NAN」が生じたかどうか、あるいは入力の一つれかが「NAN」ではないかということを確認するようにしている。本実施例では、以下に説明するようなアプローチを採用している。

【0064】

図5は、本実施例で使用している数値を表現するためのワード・フォーマット500を示したものである。このワード・フォーマット500では、1つのNANビット502aと、値の符号を与える1つのビット502bと、対数を表示する12ビット502c-eを備えている。対数を表示するビットには、5ビットの整数部502dと、6ビットの小数部502eが備えられている。負の対数を表現できるようにするために、対数を表示するビットには、対数の符号ビット502cが設けられており、これは2の補数形式で表現されている。NANビット502aは計算された値に問題が生じた場合にセットされる。図5に示すワード・フォーマット500は単なる例示であり、本発明の範囲を限定するものではない。低精度であってハイ・ダイナミ

(16)

JP 2012-530966 A 2012.12.6

ック・レンジである限りにおいては、ここで説明したものととは別のバリエーションを使用することも可能である。

【0065】

図6は、図5で示した表示形式に対応する、デジタル形式のLPHDR演算ユニット408の例を示すものである。LPHDR演算ユニット408は、二つのインプットA602aおよびB602bを受け取り、一つのアウトプット602cを出力する。インプット602a-bとアウトプット602cは、例えば、図5に示す表示形式に従い、数値を表す電気信号の形態をとることができる。そしてインプット602a-bとアウトプット602cは、LPHDR演算ユニット408のコンポーネントによって、LPHDR演算ユニット408の中で送信される真の信号である。インプット602a-bとアウトプット602cの各々は、数値のビットとNANビットから成っている。LPHDR演算ユニット408は、コントロール・ユニット(CU) 106から送られてきた制御信号412a-dによって制御される。そして、この制御信号412a-dは、インプット602a-bに対してどの演算処理を実行すべきかを決定するものである。本実施例では、インプット602a-bに対し加減算器604、乗算器606、および除算器608によって、全ての適用可能な演算処理が並行に実行される。加減算器604は、LPHDR加算および減算を実行し、乗算器606は、LPHDR乗算を実行し、除算器608は、LPHDR除算を実行する。

【0066】

所望する結果が(加減算器604、乗算器606、および除算器608の出力から)マルチプレクサ(MUXes)610aおよび610bによって選択される。右側のマルチプレクサ(MUX)610bは、アウトプット602cへ所望する値を送る。左側のマルチプレクサ(MUX)610aは、所望する処理から得られたNANビットの情報をORゲート612に送り、このORゲートは、いずれかのインプットがNANである場合や、あるいは特定の演算処理においてNANが生じた場合にはNANを設定して出力するようになっている。コンピュータのアーキテクチャを論じた文献には、図6に示す実施例に取り込むことができる多くのバリエーションが論じられている。

【0067】

対数システム(LNS)での演算では、乗算(MUL)および除算(DIV)が極めて容易であって、物理的なリソースをあまり使用しない(シリコン上での実装エリアが少ない)という、極めて大きな利点がある。演算結果の符号は、演算対象の符号の排他的論理和となる。アウトプットの対数部分は、乗算(MUL)の場合には演算対象の対数部分の和であり、除算(DIV)の場合には演算対象の対数部分の差となる。対数の和および差は、オーバーフローし、NANが生じることがある。対数システム(LNS)での演算では、他の処理も、容易になる。例えば、2乗根では、対数を1/2に除することに相当し、このことは上述した表現方法では、単に1ビットだけ右側に位置をずらすだけで良い。

【0068】

従って、図6に示す乗算器606、および除算器608は、2進法補数である入力を単に加算し、減算する回路として実装されることになる(2進法補数は、たまたま対数となっているものである)。もし、数値がオーバーフローすれば、NANとして1を出力するようになっている。

【0069】

対数システム(LNS)での加算および減算の実行、すなわち、図6における加減算器604は、対数システム(LNS)の文献にて使用されている共通的なアプローチに従って行われる。加算について考えてみる。対数bおよびcで表される二つの正の数BおよびCについて考えると、BとCの和は、 $\log(B+C)$ で表される。この結果を計算するアプローチは、当業者にとって良く知られた方法で、以下のような表示に基づく。

$$\log(B+C) = \log(B * (1+C/B)) = \log(B) + \log(1+C/B) = b + F(c-b)$$

ここで、 $F(x) = \log(1+2^x)$

従って、本実施例では、当業者に良く知られた標準的なデジタル技術を使用することにより、c-bを計算し、Fに代入し、その結果をbに加えている。



(17)

JP 2012-530966 A 2012.12.6

## 【0070】

対数システム(LNS)に関する出版された文献の多くは、 $F(x)$ 、加算(ADD)のための特別な関数、更に減算(SUB)のための類似の関数、をいかに計算するかというものに関係している。これら二つの関数は、しばしば回路を共有している。そして、このことが、図6に示す実施例において、単一の組み合わせされた加減算器604が使用されている理由である。これらの関数を計算する、あるいはその概算値を求める方法について記載した文献が多く出版されている。そして、値の精度が低い場合に、これをどのように取り扱うかについても論じられている。そのような手法、あるいはこれとは別の手法であっても使用することができる。一般論として、大規模並列LPHDR演算を実行するための、更に適切なアプローチは、回路のエリアのようなリソースの使用を最小に抑え、図6に示す実施例において使用されている表示は低精度のものであり、そこで行われる演算処理は、決定論的(確定的に1つに決定されること)なものである必要はなく、低精度の表示において最も正確な答えを返す必要もないという事実から派生する利点を享受できる。従って、本発明の実施例においては、最良の答えを計算しない回路であって、低精度の表示において適用可能な限定された選択肢から選択して使うことができる。

## 【0071】

選択した処理要素(PEs)の条件付き処理を可能にするために、本実施例では、計算結果に基づき、マスク・フラッグ410をリセットしたりセットしたりすることができるようにになっている。これを実行するメカニズムは、コントロール・ユニット(CU) 106が、データ・パス402fを通してマスク410に入ってくる値に対して、基本的なテストを行うための命令と共に、各処理要素(PE)におけるマスク410のフラッグをリセットしたり、セットしたりする命令を出し、フラッグをセットする。後述した命令の例には、データ・パス402fを通してマスク・ビット410に入るデータの符号ビットあるいはNANビットをコピーする命令が含まれている。他の例としては、データ・パス402f上のデータの12ビットの数値部が2進数のゼロに等しい場合に、マスク・ビット410がセットされるようになっている。従来のプロセッサにおける比較命令に類似したものであって、当業者に周知な上述したような処理を行う方法としては、多くの追加的な手法や別の手法が存在している。

## 【0072】

上述した対数システム(LNS)の処理を使用した自明な手法で、LPHDR演算を実行しているが、プログラマーは選択した数値を12ビットの2進法補数にすることを考慮しても良い。

乗算(MUL)と除算(DIV)は、対数システム(LNS)の演算を実行する際の演算結果の正確さの理由から、このような数値を加減算するために使用することができる。マスクのセット命令はこれらの単純な2進数を比較することができる。従って、LPHDRの計算を行う他に、対数システム(LNS)を使用したこのデジタル方式の実施例では、短い符号のついた整数に対して単純な2進数演算を実行することができる。

## 【0073】

本発明のある実施例では、アナログ方式の表示およびアナログ処理方式を備えることができる。かかる実施例では、例えば、電荷、電流、電圧、周波数、パルス幅、パルス密度、スパイクの種々の形態、あるいは従来のデジタル形式の実装の特性ではないその他の形態としてLPHDR値を表すことができる。このように表現された値の処理のためのメカニズムと同様、かかる表示方式については多くの文献により論じられている。アナログ方式と呼ばれるこのような方式は、ここで論じてきた、例えば一例としてのSIMDのような、幅広いレンジのアーキテクチャーで、LPHDR演算を実行するために使用することができる。

## 【0074】

アナログSIMDアーキテクチャーの例としては、DudekのSCAMP設計(およびその関連設計)がある。この設計では、数値はロー・ダイナミック・レンジを有し、ほぼ1%以内の精度を有している。数値は、キャパシタの電荷によって表わされるようになっている。これらのキャパシタは、典型的にはトランジスタのゲートである。各処理要素(PE)は、図4に示すレジスターに類似した数値のメモリー・セルを備えている。加算は、2つ

(18)

JP 2012-530966 A 2012.12.6

の演算対象からのバス・トランジスタをオン状態にすることによって実行され、これらの2つの演算対象は、自らの電荷をアナログ・バスに移送し、電荷と配線による自然な物理現象によって合計され、別のレジスタのキャパシタを充電するために回路を開いて移送され、演算対象の合計値が表示されることになる。Dudekによって開示された詳細なメカニズムは、実際には、負の合計値を生み出すことになるが、基本的な概念はここで述べたとおりであり、アナログ表示と単純な処理メカニズムを使い、単純な方法で加減算を実行するようになっている。

【0075】

SCAMP設計アーキテクチャーのバリエーションが製造され、そして画像処理に関連する分野において低精度、ロー・ダイナミック・レンジの処理を実行するために使用されてきている。このSCAMP設計アーキテクチャーでは、ハイ・ダイナミック・レンジの演算を行うものではなく、また、レジスタに保存された値の乗算や除算を行うメカニズムを備えているものでもない。しかしながら、Dudek設計アーキテクチャーは、アナログ方式のSIMDマシンを構築することができる可能性を示唆するものである。以下に、本発明の一つの実施例であって、LPHDR演算を実行するアナログ方式のSIMDマシンの構築方法について説明する。

【0076】

本発明の一実施例では、数値をアナログ形式とデジタル形式を組み合わせた混合方式として表示する。この実施例では、数値を、低精度で、正規化し、低を2とする浮動小数点数として表し、仮数部をアナログ値とし、指数部を2進数のデジタル値としている。アナログ値は約1%の精度であり、Dudekのアプローチに基づいており、精度はアナログ処理技術のリーズナブルな範囲内に十分入っている。指数部は、6ビットの長さであるか、あるいは要求されるハイ・ダイナミック・レンジを提供するために必要な長さであればよい。

【0077】

数値を乗ずるため、本実施例は、従来の浮動小数点法に対するアナロジーによって実行される。デジタル値である指数部は、標準的なデジタル技術である、2進数演算による加算器を使用して合計される。アナログ値である仮数部は乗じられる。これらは、約1/2と1の間の正規化した数値を表すので、これらの積は、約1/4の小さい数値となる。かかる積の値は、1/2から1の範囲に正規化し直す必要がある。本実施例では、これは、スレッシュホールド回路を使用し、アナログ値である仮数部を1/2のアナログ表示と比較することによって行われる。もし、アナログ値である仮数部が1/2より下の場合には、仮数部は2倍され、デジタル値である指数部から1を減じる。そして、この減算は、単純なデジタル値の減算である。アナログ値である仮数部を2倍にすることは、選択されたアナログ表示に対応した方法で実行される。例えば、2つのアナログ値を加算するために使用される手段であれば何であっても、コピーした仮数部を仮数部に加えることによって、仮数部を2倍にするために使用することができる。例えば、アナログ値である仮数部が電流として表現されている場合には、電流ミラーやその他の適切なメカニズムを使用してコピーを作り出すことができ、加算は、電流加算用の接点を設けることにより実行することができる。

【0078】

オリジナルのアナログ値である仮数部を乗ずる手段は、選択された表示方法に依存する。例えば、アナログ値である仮数部が、SCAMPに基づき、電荷によって表されている場合には、電荷を電流に変換するために、文献により公知となっている方法を使用することができる。例えば、キャパシタの電荷は、キャパシタの電圧を決定することになるので、電荷を電流に変換することは、電圧から電流への変換として実行することができる。そして、この技術は、アナログの電子技術の分野においては、当業者に周知となっている基本的な技術である。いずれの場合であっても、アナログ値である仮数部が電流として表示されているか、あるいは仮数部が一旦電流に変換されていれば、例えば、Gilbertの技術を使用して仮数部の乗算を実行することができる。Gilbertの乗算器は、積を表

(19)

JP 2012-530966 A 2012.12.6

す電流を作り出し、もし必要であれば、その積である電流を電荷(あるいは使用されている表示であれば何であっても良い)に変換し直すこともできる。ここで述べたことは、どのようにして処理が実行される必要があるかを示すための単なる例示である。文献では、このようなことが大々的に論じられており、この種のアナログ回路は当業者にとって周知となっている。

【0079】

数値の加減算では、従来のデジタル方式の浮動小数点演算において行われていたように、同じ指数部を有するように数値を予め正規化しておく必要がある。本実施例では、指数部を比較し、指数部が小さい方を選択することによって正規化が実行される。小さい方の指数部は、デジタル方式の手段を使用して、大きい方の指数部から減算される。この差は、小さい指数部を持つ数値の仮数部を、1/2に除することを何回行う必要があるかを特定するものである。もし、その仮数部が電流で表示されて(あるいは電流に変換されて)いる場合には、上述したように計算された指数部の差によって特定されるラダーの階数分に相当する、必要な回数だけ電流を1/2に除算するために、アナログ形式のR-2R形式のラダーを使用することができる。出力の仮数部を作り出すために、大きい方の指数部を有する数値の仮数部に対応する電流に、結果としてスケール・ダウンされた電流が加算される(もし、これがLPHDR減算処理であるならば減算されることになる)。出力される仮数部に対応する出力される指数部は、大きい方の指数部となる。この時点において、正規化の後処理が必要になる。もし、出力される仮数部が1よりも大きい場合には、仮数部を1/2に除算し、出力される指数部には1を加える(インクリメント)必要がある。もし、出力される仮数部が1/2よりも小さい場合には、1/2を超えるようになるまで仮数部を必要回数だけ、2倍に乗算する必要がある。そして、出力される指数部は、対応する回数に相当する数だけ減算(デクリメント)しなければならない。そして、このような処理は、一連のスレッシュホールド回路、倍化回路、およびデクリメント回路によって実行される。2進数デジタル形式の指数部をインクリメントしたりデクリメントしたりすることや、これらに対応してアナログ形式の仮数部の電流を2倍化したり、1/2に半減させたりすることは、当業者に良く知られた単純な処理である。

【0080】

本実施例では、指数部はデジタル形式の2進数として表されている。別の実施例では、指数部をアナログ値として表すこともできる。しかし、指数部は保存時および計算時においては、指数部を表す数値に変動を起こすようなノイズも誤差も生じさせることのない方法で表されることは、特に重要な点である。指数部におけるかかる変動は、記憶された数値の値に、2つの変化するファクターを生み出すことになる。指数部の精度を維持するための一つの実施例として、指数部を比較的小さいレベル、例えば、16ビット+符号ビットに量子化することができる。処理途中において、指数部のアナログ表示部に生じたわずかな変動は、16ビットの標準化された量子化レベルに数値を記憶し直すようになっている回路によって、取り除くことができる。このような実施例において、十分なダイナミック・レンジを得るために、浮動小数点数は、通常の低2の数値としてではなく、低4の数値として処理するようにしても良い。例えば、このことは正規化された仮数部は1/4から1の範囲のレンジにあることを意味する。ここで述べた加算、減算、乗算はわず

【0081】

上述したアナログ形式で混合した信号を使用した実施例は、単に例示するものであって、本発明の限定を構成するためのものではない。神経形態学的であって、アナログ形式で混合した信号を使用した技術に関する文献刊行物は、LPHDR演算において実行することができる記憶と処理に関する手法について、非常に多くのものを提供している。このような記憶と処理においては、LPHDR演算を実行するマシンの動作に、組立誤差と同様にノイズも取り込まれてしまう。「fp+ノイズ」演算を使用したソフトウェア・アプリケーションの実行において得られる後述するような結果は、非常に「非デジタル」的なクオリティを持つものであるにもかかわらず、このように構築されたマシンは驚くほど有用な



ものである。

【0082】

LPHDR演算がいくつかの重要な計算アプリケーションの実務において、有効である証拠について以下にのべる。本発明の実施例の幅広いバリエーションに対して、有効であるという証拠が存在するが、この有効さは、実施内容の細部に依存するものではない。

【0083】

LPHDR演算の有効性を示す目的のために、LPHDRマシンの非常に一般的な例を選択する。

かかるマシンのモデルは、少なくとも以下のような機能を定提供できるものである。

(1)大規模な並列処理ができ、(2)ノイズを伴う可能性のあるLPHDR演算を提供でき、(3)各演算ユニットに少量の局所メモリを備えており、(4)ユニット間に局所接続(パワフルで、フレキシブルで、あるいは高度な接続メカニズムではない接続)のみを備えた2次元的な物理的配置を有する演算/記憶ユニットを備えており、(5)マシンとホスト・マシンの間には、限定されたバンド幅のみを備えているマシンである。

10

なお、このモデルは、本発明の実施例に係るバリエーションの有効性を論証するためにだけ例示するものであり、本発明を限定するためのものではないことに留意すべきである。このモデルは、特に、デジタル形式、アナログ形式、あるいはこれらを複合したもので実行し、ノイズがゼロまたはノイズを有するものであり、FPGA、SIMD、MIMDのようなアーキテクチャーやモデルの前提条件に適合したアーキテクチャーを備えている。そして、共有メモリ設計、GPUのような設計、あるいはその他の洗練された設計などの、一般的なアーキテクチャーは、このモデルの能力を組み込んでいる。そのため、これらのアーキテクチャーを使ったLPHDR演算は、更に有用なものとなっている。ここでは、LPHDR演算は広い範囲の設計に対して有効であることを示しているが、広い範囲の設計の中のSIMDは、以下のディスカッションのための一つの例として示すものであり、各ユニットを「処理要素」又は「PE」と呼んでいる。そして、各ユニットはメモリーと演算を組み合わせ

20

【0084】

数種類のアプリケーションについて以下に説明する。各アプリケーションに対し、以下の点について論じている。すなわち、(1)ノイズの発生の可能性のあるLPHDR演算において計算が実行されるとき、その結果は有効なものとなること、(2)各ユニット間でのデータの局所的なやり取りのみであり、各ユニット内にあるメモリーに限定し、そしてホスト・マシンとの間でのデータのやり取りに限定して、2次元的な物理的配置で計算を構成することにより、マシンのリソース(領域、時間、電力)を効率的に使用した計算となる。

30

第1の要求事項は、「精度」に関する事項あり、第2の要求事項は、「効率」に関する事項である。この二つの要求事項を満たし、このモデルで動作するアプリケーションは、多くの種類のLPHDRマシン上で十分に機能するものと考えられ、従って、このような本発明のマシンは幅広く有効活用できるものである。

【0085】

アプリケーションは、2つの実施例にかかるマシンの演算によってテストされた。1番目の実施例は、正確な浮動小数点演算を行うものであるが、各演算処理の結果に0.99から1.01の間のランダムな数値であって、一律に選択した数値を乗ずるようになっている。以下の議論において、この実施例を「fp+ノイズ」と表示することにする。この実施例は、アナログ形式の実施例にかかるマシンによって算出された結果を表すものである。

40

【0086】

2番目の実施例は、図5に示す表示形式の数値により、対数演算を使用するものである。

演算は繰り返しが可能、すなわちノイズを含まないが、小数部のサイズが小さいため、各処理において約1~2%までの誤差が生じる。以下の議論において、この実施例を「lns」と表示することにする。この実施例は、特に、デジタル形式の実施例にかかるマシンによって算出された結果を表すものである。

【0087】

本発明にかかる実施例の有用性を論証するために、本発明に係る実施例によって実行可

50

能であり、実務的なアプリケーションの多様化が可能である、3つの計算タスクについて論じることとする。タスクの内の2つは、最近傍探索に関連するものであり、残りの1つは、視覚情報を処理することに関連するものである。以下に、タスクについて説明した後、その実務的なアプリケーションについて述べ、その後で、各タスクが上述した一般的なモデルを使用して計算することができ、かつ本発明に係る実施例を適用して計算することができることを論証する。

#### アプリケーション1: 最近傍探索

##### 【0088】

大きなセットのベクトル群が与えられており、これを「Examples」と呼ぶことにする。そして、ある一つのベクトルを「Test」と呼ぶことにする。最近傍探索という課題(「NN」と呼ぶ)は、「Test」に最も近い「Example」を探し出すことであり、ここで計量的な距離は、ユークリッド距離の2乗(各コンポーネント間の距離の2乗和)とする。

##### 【0089】

NNは、幅広く使用される有用な計算である。一つの応用例として、データ圧縮の分野があり、「ベクトル量子化」と呼ばれている。このアプリケーションにおいては、「コード・ブック」内において、短い「コード・ワード」に関連づけられた、比較的長いベクトルのセットを準備する(これらがExamplesである)。圧縮すべき一連のベクトル上を移動し、そのうえでTestであるベクトルに対して、コード・ブック内で最も近いベクトルを見つけだし、これに対応するコード・ワードを出力する。これによって、一連のベクトルは、短い連続したコード・ワードに減縮される。コード・ワードは、オリジナルの一連のベクトルを完全に特定するものではないので、このやり方は、不可逆形態のデータ圧縮になる。その他のアプリケーションとして、音声の圧縮やMPEG標準として利用することができる。

##### 【0090】

NNの別のアプリケーションとしては、大きなビデオ・データベースにおいてビデオの断片化が起こるか否かを決定するために利用することができる。ここでは、色ヒストグラムや、スケール不変の特徴抽出等の既知の手法を使って、ビデオの断片からの画像を要約して特徴ベクトルにしている。ここでExamplesは、ビデオのデータベースから抽出された類似の特徴ベクトルとなる。ビデオの断片からのいずれかのベクトルがビデオのデータベースからのいずれかのベクトルに近かったかどうかについて知ることを希望しており、このような判断を行うに当たってNNは有効になる。

##### 【0091】

最近傍探索に関するアプリケーションの多くにおいて、真に最も近隣のものを見つけ出すことが好ましいが、もし、ほとんどの場合真に最も近隣のものを見つけ出し、たまにわずかに離れた別の近隣のものを見つけ出すような場合であっても、それは許容できるものである。従って、最近傍探索の課題の近似解も有用であり、それが特別に速く、あるいは低電力で、あるいは厳密解を得る場合に比べ何か別の利点を持って計算できるものであるならば、特に有効である。

##### 【0092】

以下に、「精度」と「効率」に関するクライテリアを満たす方法であって、本発明の実施例を適用して、ほぼ最も近隣のものを見つけ出すことが可能であることを示す。

##### 【0093】

#### アルゴリズム

本発明の実施例に従い、実装されたマシンによって実行することができるアルゴリズムについて以下に説明する。そして、アルゴリズムを実行するための命令を含む実行ソフトウェアによって、アルゴリズムは実行される。このアルゴリズムへの入力は、1セットのExamplesとTestベクトルである。そして、このアルゴリズムは、Testに最も近い(あるいはほぼ最も近い)Exampleを見つけ出すものである。

(22)

JP 2012-530966 A 2012.12.6

【0094】

最もシンプルなバージョンのアルゴリズムでは、Examplesの数は処理要素(PEs)の数を上回らず、各ベクトルは単一の処理要素(PE)のメモリー内に収まる程度の小さい情報量でなければならない。各処理要素(PE)の中には一つのExampleが配置されるようにして、Examplesが処理要素(PEs)のメモリー内に配置される。Testが与えられると、Testは全ての処理要素(PEs)を順に通過するようになっている。Testは処理要素(PEs)を通過するようになっているので、Testに同伴することによって、どの処理要素(PE)が(従って、どのExampleが)最も近いExampleであるかが分かると共に、Testから最も近いExampleまでの距離が分かる。各処理要素(PE)は、Testと処理要素(PE)のメモリーに記憶されたExampleとの間の距離を計算し、距離とこのPEの中へ渡されたインジケータと一緒にTestを転送するか(このPEによって計算された距離が、このPEへ渡された距離を超えている場合)、あるいは、このPEのExampleが最も近いものであることを示す情報と共に、このPEによって計算された距離と一緒にTestを転送する(このPEによって計算された距離が、このPEへ渡された距離よりも小さい場合)。従って、このアルゴリズムでは、Testは処理要素(PEs)のセットを通過していくので、シンプルな最小化された処理が行われることになる。Testと関連情報が最後のPEを出ていくとき、出力にはExampleとTestの間の距離と共に、どのPE(そしてどのExample)がTestに最も近いかという表示が含まれている。

10

【0095】

このアルゴリズムの更に効率的な改良タイプのもものでは、Testは最初に転送され、例えば、一番上の行から列の全てへとTestと関連情報を下方に送り、他の列と並行して効率的なサーチが行われる。そして、一旦情報が列の下端まで届くと、列の下端を横断して情報が伝達される限り、処理される全ての列のExampleの最小の距離を計算する。このことは、Testを処理するために必要となる時間が、行および列における処理要素(PEs)の数に(大きく依存して)比例することを意味する。

20

【0096】

このアルゴリズムの機能強化は上述したように行われるが、機能強化されたアルゴリズムは、見つけ出された最も近くのExampleおよび2番目に近くのExampleの両方を示す情報を計算し、そして伝達する。この情報が処理要素(PEs)のアレイから出ていくとき、処理要素(PE)アレイを管理するデジタル・プロセッサがTestと、処理要素(PE)アレイによって示された2つのExamplesの間の距離を(高精度で)計算し、2つの内の最も近いものが、Testに最も近い近隣のもののようだとし、出力される。

30

【0097】

#### 精度

C言語でコーディングされ、上述したような機能強化したアルゴリズムによって実行された演算について説明する。このコードでは、後で述べるような重みづけしたスコアと共に、上述した最も近い近隣のものを計算する。

【0098】

Cコードは、ハードウェアで実行されたように、本発明を実際に実行するときに行う演算と同じ方法を使って、同じオーダーの、同一セットの演算処理を行うものである。従って、Cコードは、本発明を実行する際に機能強化したアルゴリズムが導き出す結果と同じ結果を出すことができるものである。(このように実行する際に、効率的に機能させるために、アルゴリズムはどのように体系づけられているかということについては、「効率」について論じるセクションにおいて説明する。)

40

【0099】

特に、Testと各Exampleの間の距離を計算する際、TestとExampleのベクトル成分間の距離の二乗の和を求めるために必要となる計算において、長い数値の和を求めることになる可能性があるため、Cコードは、後述するようにKahanの方法を使うようにしている。

【0100】

このCコードは、上述したように、演算のための数種類の実行手続きを含むようになっている。「#define fp」でコンパイルすると、IEEE標準浮動小数点を使用して演算が

50

(23)

JP 2012-530966 A 2012.12.6

行われる。もし、ノイズ下での演算を可能にするために、コマンドライン引数が出されたならば、各計算結果にランダム・ノイズが加えられることになる。これが「fp+ノイズ」形式の演算である。「#define fp」無しでコンパイルした場合には、低が2であって6ビットの小数部を有する低精度の対数演算を使用することによって、演算が実行される。これが「lms」形式の演算である。

【0101】

このコードが実行されると、コードが実行した計算結果を示す出力が作り出される。これらの出力は、以下に説明するが、あるコマンドライン引数によって、機能強化されたアルゴリズムがLPHDRの最も近隣ののものに関する計算結果を生み出すことを示している。

これらの結果は、このアプローチが有用なものであることを示すものである。ここでは、これらの結果について簡単に説明する。

【0102】

第1番目の結果は、「fp+ノイズ」形式に関するものである。10回の明確に区別できる計算を実行した。各計算の実行においては、百万のランダムなExampleベクトルであって、長さ5のExampleベクトルを発生させた。そして、各ベクトルの各コンポーネントは $N(0, 1)$ に基づき決定されたものである。ここで、 $N(0, 1)$ は、ガウス(正規)分布であって、平均がゼロ、標準偏差が1を意味する。更に、各計算の実行においては、百のTestベクトルであって、長さ5のTestベクトルを発生させた。そして、各ベクトルの各コンポーネントは同じく $N(0, 1)$ に基づき決定されたものである。各Testに対して、2つの方法により最も近隣のものを計算した。一つは機能強化した上述のアルゴリズムを使用するものであり、もう一つは、高精度で浮動小数点演算による標準の「最近傍探索方法」により計算するものである。機能強化したアルゴリズムを使用したものは、標準の浮動小数点法をしたものと同じ結果となった。その結果を以下に示す。

```
% ./a.out 5 10 1000000 100 1
```

表現形式は浮動小数点形式であり、ノイズ有

```
Run 1: 100 Testに対し実行、100(100.0%)適合性、 0.81%平均スコア誤差
Run 2: 100 Testに対し実行、100(100.0%)適合性、 0.84%平均スコア誤差
Run 3: 100 Testに対し実行、100(100.0%)適合性、 0.98%平均スコア誤差
Run 4: 100 Testに対し実行、100(100.0%)適合性、 0.81%平均スコア誤差
Run 5: 100 Testに対し実行、100(100.0%)適合性、 0.94%平均スコア誤差
Run 6: 100 Testに対し実行、100(100.0%)適合性、 0.82%平均スコア誤差
Run 7: 100 Testに対し実行、100(100.0%)適合性、 0.78%平均スコア誤差
Run 8: 100 Testに対し実行、100(100.0%)適合性、 0.86%平均スコア誤差
Run 9: 100 Testに対し実行、100(100.0%)適合性、 0.85%平均スコア誤差
Run 10: 100 Testに対し実行、99(99.0%)適合性、 0.86%平均スコア誤差
```

ここで、LPHDR演算が最も近いExampleを見つけ出した回数の平均パーセント(最終的にDPによる修正を行ったもの)=99.90%

LPHDR演算とBPとの対比による平均スコア誤差の平均値=0.85%

【0103】

ここで、「平均スコア誤差」の値については、後で述べる「重みづけスコア」に関する議論の部分において説明する。「適合性」という情報は、ここでは関連性のあるもの(適切であるもの)を意味する。

【0104】

10回の計算の実行、計算の実行に際してはいずれかのTestのみを対象にし、100個のTestについて計算を実行した結果得られた最も近隣のものは、通常の高精度な手法により得られた最も近隣ののものとは違った結果が得られた。従って、「fp+ノイズ」形式の演算であって、機能強化したアルゴリズムにより実行した場合と、通常の高精度な手法により実行した場合とにおける適合性を対比した平均パーセントは99.9%であった。

(24)

JP 2012-530966 A 2012.12.6

## 【0105】

次に、「lms」形式の演算を使用して類似の計算を実行した。この結果を、以下に示す。

```
% ./a.out 5 10 1000000 100 0
```

表現形式はlms形式であり、ノイズ無

```
Run 1: 100 Testに対し実行、100(100.0%)適合性, 0.15%平均スコア誤差
Run 2: 100 Testに対し実行、100(100.0%)適合性, 0.07%平均スコア誤差
Run 3: 100 Testに対し実行、100(100.0%)適合性, 0.08%平均スコア誤差
Run 4: 100 Testに対し実行、100(100.0%)適合性, 0.09%平均スコア誤差
Run 5: 100 Testに対し実行、100(100.0%)適合性, 0.11%平均スコア誤差
Run 6: 100 Testに対し実行、100(100.0%)適合性, 0.16%平均スコア誤差
Run 7: 100 Testに対し実行、100(100.0%)適合性, 0.07%平均スコア誤差
Run 8: 100 Testに対し実行、100(100.0%)適合性, 0.13%平均スコア誤差
Run 9: 100 Testに対し実行、99(99.0%)適合性, 0.17%平均スコア誤差
Run 10: 100 Testに対し実行、98(98.0%)適合性, 0.16%平均スコア誤差
```

ここで、LPHDR演算が最も近いExampleを見つけ出した回数の平均パーセント(最終的にDPによる修正を行ったもの)=99.70%

LPHDR演算とDPとの対比による平均スコア誤差の平均値=0.12%

## 【0106】

この結果では、適合性を対比した平均パーセントは99.7%であり、「fp+ノイズ」形式のものよりわずかに悪い結果となった。

## 【0107】

LPHDR演算の2つの形式を使用した、最近傍探索の機能強化したアルゴリズムによって示された精度は、驚くべきものであった。1%の誤差で多くの計算を連続的にを行い、しかも、1%以下の誤差で最終的な計算結果を導き出すということは、これまでの経験からは考えられないことである。それにもかかわらず、LPHDR演算が効率的であることを立証し、ほぼ最も近い近隣のものを見つけ出すという計算方法が有効な応用分野におけるアプリケーションにおいて、得られた精度は十分高く、有用なものであった。

## 【0108】

極端なケースとして、「fp+ノイズ」形式の変形方式について評価を行った。この場合には、ノイズが+10%から-5%までの間を滑らかに変化するようにしている。従って、各演算処理によって、10%の非常に大きいものから5%の非常に小さい範囲の結果が得られた。上述した、最近傍探索の機能強化したアルゴリズムを実行した。ここで、各計算の実行においては、10万のExampleベクトルを発生させた。その結果、以下のような、不明確で、ノイズが存在し、非ゼロ平均のLPHDR演算という極端な条件であっても、有用な結果を実現することができるという、驚くべき結果が得られた。

```
Run 1: 100 Testに対し実行、97(97.0%)適合性
Run 2: 100 Testに対し実行、100(100.0%)適合性
Run 3: 100 Testに対し実行、100(100.0%)適合性
Run 4: 100 Testに対し実行、98(98.0%)適合性
Run 5: 100 Testに対し実行、98(98.0%)適合性
Run 6: 100 Testに対し実行、99(99.0%)適合性
Run 7: 100 Testに対し実行、99(99.0%)適合性
Run 8: 100 Testに対し実行、99(99.0%)適合性
Run 9: 100 Testに対し実行、99(99.0%)適合性
Run 10: 100 Testに対し実行、99(99.0%)適合性
```

ここで、LPHDR演算が最も近いExampleを見つけ出した回数の平均パーセント(最終的にDP



(25)

JP 2012-530966 A 2012.12.6

による修正を行ったもの)=98.80%

### 【0109】

#### 効率

驚くべき精度の結果に対し、ここで示した計算モデルにおいて、最近傍探索の機能強化したアルゴリズムによる計算が効率良く実行されたということは、この技術分野における通常の知識を有する者にとっては明らかなことである。ここで、マシンの演算/メモリのユニットは2次元的な物理的配置で接続されており、処理要素(PEs)間のローカルな通信のみを使用したものである。しかし、このことは、ホスト・マシンに対して狭いバンド幅を使い、有用な作業を行わせて、マシンを使用中の状態に維持することを狙ったものではない。

10

### 【0110】

単一のTestに対して最近傍探索の計算を行うとき、Testはアレイ中の全ての処理要素(PEs)を通過していく。上述したように、アレイが $M \times M$ の格子状であれば、そのTestがマシンを通過し、結果をホスト・マシンに返すために、当該Testに対して少なくとも $O(M)$ ステップの演算が行なわれる。この間、マシンは最も近隣なものとの間の距離の計算を $O(M \times M)$ ステップ実施することになる。しかし、マシンは各ステップにおいて、 $O(M \times M)$ ステップの計算実行する能力があるので、 $O(M)$ ステップのファクターが失われることになる。

### 【0111】

シリアルタイプのマシンに比べ、このようなマシンはスピード・アップが図れ、 $O(M)$ ステップのファクターは重要で有用なものとなる。しかも、効率は更に高くなる。もし、十分に多くのTestベクトル、いわゆる $O(M)$ ステップ、あるいは更に多くのTestベクトルを処理しなければならなくなると、それらはマシンの中に流れ込み、パイプライン状に通過する流れを構成することになる。 $O(M)$ ステップのTestを処理する時間は、 $O(M)$ ステップのままであり、単一のTestに対しての時間と同じである。しかし、ここでマシンは、 $O(M) \times O(M \times M)$ ステップの距離計算を実行することになり、その結果、一定のファクターではあるが、マシンの計算能力の全てが使用されることになる。

20

### 【0112】

もしマシンが、処理要素(PEs)の数の2乗根と同じ数のTestベクトルを処理するのであればマシンは特に効率的になる。このような形態によく適合するアプリケーションが存在する。このようなアプリケーションとして、Examplesのセット中の各Exampleに最も近隣のものを見つけ出す問題と同様に、パターン認識や、多数の独立したTestの圧縮に関するもの(例えば、画像のブロック、ファイルの部分、独立した株の株価履歴など)がある。

30

ここで述べたことは、上述したように、いわゆる当業者の観点に反するものであり、単一のチップ上に非常に多くの演算処理要素を配置したマシンや、これに類似するものは、有用なものではない。

## アプリケーション2: 距離の重みづけスコアリング

### 【0113】

40

最近傍探索の問題に関連するタスクとして距離の重みづけスコアリングがある。このタスクでは、各Exampleは関連づけられたスコアを有している。このスコアは、Exampleをいずれかの方法により特徴づける数値である。例えば、Examplesが与えられた株の株価履歴を抽出したものであれば、スコアは、株価がほぼ上がりそうか、下がりそうかの見込の履歴になるかもしれない。Testベクトルが与えられると、タスクは、全てのExamplesのスコアの重みづけした和を求めることになる。ここで、重みは、Testから各々のExamplesまでの距離の漸減関数となる。例えば、この重みづけされたスコアは、Testによって履歴が表された株の将来の株価の予測値として取り扱われるかもしれない。本発明の実施例をこのように適用することにより、例えば、量的なヘッジ・ファンドによって行われているような株の高速トレーディングをサポートすることができるようになるかもし

50



(26)

JP 2012-530966 A 2012.12.6

れない。そして、このことは、当業者の観点である、低精度の計算はフィナンシャルの分野でのアプリケーションとしては有効ではないとする考え方に反するものである。

【0114】

上述したCコードにより、最近傍探索の計算と共に、重みづけしたスコアが計算される。この計算においてExamplesに割り当てられたスコアは、範囲[0, 1]から引き出されたランダムな数値である。この計算における各Exampleに対する重みは、全てのExamplesに対する非正規化された重みの和によって除算されたExampleに対する非正規化された重みであると定義される。

そして、各Exampleに対する非正規化された重みは、ExampleからTestベクトルまでの距離の2乗プラス1の和の逆数であると定義されている。上述したように、Cコードは、多数の計算を実行し、その計算において多くのExamplesとTestsを導き出し、そして、「fp+ノイズ」形式および「lms」形式を使用して計算された結果と従来の浮動小数点による計算結果とを比較した。

10

【0115】

上述したシミュレーションを実行した出力結果を再び参照すると、「fp+ノイズ」形式によるLPHDR演算における重みづけスコアの平均値は、修正した値の0.85%以内にあり、1%を超えるものではなかった。「lms」形式による演算においては、誤差は非常に小さく、平均値でちょうど0.12%の誤差であった。

【0116】

これらの結果は驚くべきものであり、重みづけされたスコアの計算には、各Exampleに関係する個々の重みづけされたスコアの合計がふくまれているということを示している。

20

各計算の実行において、百万のExamplesが処理されているので、このことは、合計値は百万以上の小さい正の値であった。各加算において約1%の誤差をもって、百万の小さな値の合計値を求める単純化した方法は、概算によるノイズを含んだ結果を生み出すことになる。しかし、このCコードは、Kahan(Kahan, William(January 1965),「切り捨て誤差を低減する際の注意点」ACM8(1)のコミュニケーション:40)によって発明され、長い間公知となっていた方法を使用して合計値を求めるものである。この方法は、距離の重みづけスコアを求める際、あるいは、モンテ・カルロ法を使ったデリバティブ証券の価格計算を行うなどの金融工学の分野、あるいは、次に説明する画像処理アルゴリズムでのデコンボリューションを実行する際に使うこともできるような、長い加算を行うことができようにするものである。

30

【0117】

このアルゴリズムの効率は、最初のほうで述べたように、NNのアルゴリズムの場合と類似している。もし、一回に多くのTestベクトルが処理されるような場合には、マシンは特に効率よくタスクを実行する。

### アプリケーション3: 画像の動きに起因するぼやけの除去

【0118】

画像を形成する光を十分に集めるために、カメラは往々にして長時間シャッターを開けたままの状態に置かれ、このためカメラの動きによって「ぼやけ」が生じることになる。

40

人口衛星や航空機に搭載される移動用カメラは別として、非常に高価なカメラと同様安価な民生用カメラでは、カメラの揺れの結果として、このような「ぼやけ」が生じることがある。もし、カメラの移動経路が分かっているのであれば(あるいは、計算することができるのであれば)、「ぼやけ」はいろいろな「ぼやけ」除去のアルゴリズムを使用して、実質的に取り除くことができる。かかるアルゴリズムの一つとして、Richardson-Lucyの方法(「RL手法」と呼ぶ)がある。ここでは、本発明の実施例が、Richardson-Lucyの方法によるアルゴリズムを実行することができ、有用な結果を導き出すことができることを示す。上述したアルゴリズムのフォーマットの説明に続き、精度と効率に関するクワイテリアについて述べる。

50

(27)

JP 2012-530966 A 2012.12.6

## 【0119】

アルゴリズム

Richardson-Lucyの方法によるアルゴリズムはよく知られ、幅広く利用されている。この方法では、画像は既知のカーネルによって、ぼやけさせられていると仮定する。特に、このカーネルは直線であると仮定し、「ぼやけ」が純粋に水平方向にのみ生じるように画像が方向づけられているものと仮定する。ぼやけた画像の各列におけるJ番目のピクセルに対する特定のカーネルが、ぼやけていない画像のJ番目からJ+31番目のピクセルの様に重みづけした平均値であると考える。

## 【0120】

精度

Cプログラミング言語を使い、LPHDR演算を使用したRL手法の簡単なバージョンについて計算を実行した。このプログラムでは、テスト画像を読み込み、上述したようなカーネルを使って画像をぼやけさせ、そして、「fp+ノイズ」形式か「lms」形式のいずれかの演算方式を使用して画像のぼやけ除去を行った。ぼやけ除去を行った画像の現状の近似値でカーネルをコンボリューション(たたき込み積分)する際のように、RL手法のアルゴリズムによって和を計算した。この計算の実行にあたっては、最初にのべたように、Kahanの手法を使用してこれらの和を求める計算を行なった。図7は、テスト画像のオリジナルの状態を示したものである。このテスト画像は、バラク・オバマの就任式に使われた建物の衛星画像である。図8は、カーネルによって極端にぼやけさせられた画像を示すものである。この画像では、いずれの特定の対象物をも見分けることができない。図9は、標準の浮動小数点演算を使用して、ぼやけ除去を行った結果を示すものである。図10は、「fp+ノイズ」形式の演算を使用して、ぼやけ除去を行った結果を示すものである。そして、図11は、「lms」形式の演算を使用して、ぼやけ除去を行った結果を示すものである。これらの全てのケースで、画像は、建物、道路、パーキング・ロット、および車を識別することができる状態で、再保存された。

## 【0121】

人間の目を使って判定するために画像を表示するのに加え、ぼやけ除去の性能に関する数値的特性値の計算を行った。この計算は、画像内の全てのピクセルに対し、オリジナルの各ピクセルの値(0から255までのグレイ・スケール値)と、RL手法のアルゴリズムによって再現された画像の対応する各ピクセル値の間の平均差を求めたものである。結果として得られた数値的特性値を下の表1に示す。

表 1

画像のタイプ	平均ピクセル誤差
ぼやかした画像	32.0
標準浮動小数点を使った RL 手法で処理した画像	13.0
「fp+ノイズ」を使った RL 手法で処理した画像	13.8
「lms」を使った RL 手法で処理した画像	14.8

## 【0122】

これらの結果と、主観的ではあるが重要な人間の目による判定によれば、LPHDR演算により、標準浮動小数点による演算に比べても、実質的であって有用なぼやけ除去が可能であることが示された。更に、この例では、LPHDR演算を使用したぼやけ除去のビジュアルなインパクトとそのコンセプトをよりよく伝えるために、極端にぼやけた画像を選んである。もっと穏やかで、一般的なカーネルによるぼやけでは、得られるぼやけた画像は、今回のケースの画像より、もっとオリジナルの画像に近いものであり、カーネルの長さが縮小され、このような画像に対してLPHDR演算を使ったRL手法のアルゴリズムを実行す

(28)

JP 2012-530966 A 2012.12.6

ることにより、もっと一般的なものとして見るができるようになる。

【0123】

効率

局所的なカーネルを使用したRichardson-Lucyの方法は、局所的な計算処理のみを行うものであることは、いわゆる当業者には自明なことである。ぼやけ除去すべき画像は、処理要素(PE)のアレイの中へロードすることができ、処理要素(PE)毎に一つまたはそれ以上のピクセルを保存し、RL手法のアルゴリズムによるデコンボリューション処理を数十回または数百回繰り返すことができる。そして、ぼやけ除去された画像は、ホスト・プロセッサに戻され、読み込まれるようになっている。十分に長い繰り返し処理を行う限り、マシンを効率よく使用することができる。

10

【0124】

画像のぼやけ除去に関する極端な形態として、コンピュータ・トモグラフィー(断層撮影)に使用される、繰り返し復元法がある。2Dの投影画像から3Dの立体画像を復元することは、極端に大量の計算タスクを必要とするものである。前述した方法は、本来、繰り返し復元法に一般化されるものであり、マシンを効率的に使用することになるものである。

【0125】

本発明の実施例の利点としては、以下に記載するものの内の一つまたはそれ以上のものがある。

【0126】

本発明のある実施例によって実行された処理要素(PEs)は、演算を実行できる処理要素(PEs)としては比較的小さいものである。このことは、単位リソース(例えば、トランジスタ、面積、体積など)当たりに、多くの処理要素(PEs)があることを意味し、そしてこのことは、単位リソース当たりの演算にかかる計算パワーが非常に大きいことを意味する。このことによって、従来のコンピュータ設計に比べ、与えられた範囲のリソースで、より大きな問題を解決することが可能になる。例えば、最新の技術で組み立てられた大きなシリコン・チップとして構築された、本発明のデジタル方式の実施例では、1サイクル当たり、数万回の演算処理を実行することができる。これに対し、従来のGPUでは、1サイクル当たり数百回程度であり、従来のマルチ・コアCPUでは、1サイクル当たりごく少数回の演算処理しか実行できない。このような演算回数の比率は、本発明の実施例のアーキテクチャ上の利点に反映される。それは、現状ナノ・テクノロジーに到達し、あるいはデジタル方式とアナログ方式による計算の実行技術に到達しているにも関わらず、継続的に改良が進められている組立技術として存続するものである。

20

30

【0127】

一般的には、少ないリソースで演算を実行すること、および実施例の中で明確に示されていることは、演算に使用される電力が小さいということの意味している。その結果、本発明の実施例によって実行されるマシンは、リーズナブルな使用電力(例えば、数十ワット)で、極端に高い性能を発揮しているか、または少ない使用電力(例えば、何分の1かの電力)で、高い性能を発揮していると言える。このことは、かかる実施例は、スーパーコンピュータから、デスクトップ・コンピュータや、モバイル・コンピュータに至るまでのフル・レンジのコンピュータに適したシステムであることを意味する。同様に、コストは一般論として、利用できるリソースの量に関連するので、本発明の実施例は、従来のコンピュータ・デバイスに比べ、比較的大きな、単位コスト当たりの計算パワーを提供できるようになっている。

40

【0128】

SIMDアーキテクチャは、かなり古いものであり、いわゆる当業者からはコンピュータ・デザインのアプローチの手段としては放棄されたものである。しかし、一般的な演算能力のような重要な機能を維持しており、SIMDマシンのプロセッシング・エレメントが特に小さく作ることができれば、SIMDのアーキテクチャは有用なものとなる。ここで示した実施例は、まさにこのような品質を備えたものである。

50

(29)

JP 2012-530966 A 2012.12.6

## 【0129】

大規模なLPHDR演算は有用でないとする一般的な見方があったにもかかわらず、大規模なLPHDR演算は、極めて一般的な計算フレームワークとして有用であるという発見は、単にSIMDマシンの実施例のみならず、(大規模であるか否かにかかわらず)どんな並列処理マシンにおいても有効に活用することができる。大規模なLPHDR演算は、FPGAs、FPAAs、GPU/SIMTマシン、および少ないリソース(トランジスタや体積など)を使い、大量の計算を実行するためのコンパクトな演算処理要素を使ったあらゆる種類のマシンにおいて使用することができる。

## 【0130】

本発明の実施例の別の利点としては、一般論として効率的に計算を実行できるから有用であるということではなく、一般的に高精度な演算要素が必要になると考えられる、種々の現実的な問題に取り組むために使用することができるという点が挙げられる。たとえば本発明の実施例が、低精度の計算要素しか備えていない(あるいは、計算要素の大部分が低精度である)としても、以上のようなことが言える。ここで、かかる現実的な問題のいくつかの例について述べ、そして分子動力学のシュミレーションにおける非結合力場の計算やその他のタスクを実行することにも成功しているが、これらは単なる例示であって、本発明の実施例を使用することによって解決することができる、現実的な問題の範囲を限定するものではない。

## 【0131】

ここで開示した実施例は、単なる例示であって、本発明を限定するものではない。むしろ、本発明の実施例は、以下に示すような、他の種々の方法でも実行することができる。

## 【0132】

例えば、本発明の実施例は、デジタル形式やアナログ形式の表示を使ったり、固定小数点、対数、あるいは浮動小数点表示、電圧、電流、電荷、パルス幅、パルス密度、周波数、確率、スパイク、タイミング、あるいはこれらの組み合わせなど、種々の方法のいずれかの方法により値を表示することができる。このような内在する表示方法は、個別に、あるいは組み合わせによりLPHDRの値を表すために使用することができる。LPHDR演算回路は、種々のデジタル方式(パラレルかシリアルか、あるいはパイプライン型かそうでないか)、あるいはアナログ方式、あるいはこれらの組み合わせのような、種々の方法の中のいずれかの方法によって実施することができる。演算要素は、最近接した4つ、最近接した8つ、種々のタイプの飛び石状、あるいは三角形やグリッド状になったりならなかったりするアーキテクチャーのように、種々の接続アーキテクチャーを使って接続することができる。パラレルまたはシリアル、デジタルまたはアナログ、あるいはこれらを組み合わせたモードを利用した方法のように、どのような方法であっても、演算要素間の通信を行うために使用することができる。演算要素は、同期して処理したり、あるいは非同期で処理を行い、そして全体的に同時に処理したり、あるいは全体的に同時にでなく処理したりすることができる。演算要素は、例えば、シリコン・チップのような、単一の物理的デバイス、あるいは散在する複数のデバイスに対して実行することができる。そして複数のデバイスから構築された実施例では、例えば、グリッド状、トラス状、ハイパーキューブ状、ツリー状、あるいはその他の方法を含む種々の方法で接続された演算要素を備えている。演算要素は、もしあるならば、コストとバンド幅、そして特定の実施例に対するその他の要求事項に基づいて、種々の方法で、ホスト・マシンに接続することができる。例えば、演算要素の集合体に接続される多くのホスト・マシンが存在する。

## 【0133】

本発明のある実施例については、SIMDアーキテクチャーとして実施されるとして説明しているが、これは単なる例示であって、本発明をこれに限定するものではない。例えば、本発明の実施例は、プログラマブルなロジック・デバイス、フィールド・プログラマブル・アナログ・アレイ、あるいはフィールド・プログラマブル・ゲート・アレイ・アーキテクチャー(これらに限定するものではないが)のように、再構成可能なアーキテクチャー



(30)

JP 2012-530966 A 2012.12.6

として実施することもできる。そして、フィールド・プログラマブル・ゲート・アレイ・アーキテクチャーには、FPGAの既存の乗算器ブロックが、ここで開示したいいずれかのタイプのLPHDR演算要素で置き換えられるか、あるいは補完されるようにしたデザインのもの、あるいは、LPHDR演算要素が新規あるいは既存の再構成可能なデバイスの中に含まれるようにしたデザインのものがある。別の例として、本発明の実施例は、GPUあるいはSIMT形式のアーキテクチャーとして実施できる。そして、このアーキテクチャーには、ここで開示したいいずれかのタイプのLPHDR演算要素が取り入れられている。例えば、現在のあるいは新しいグラフィック・プロセッシング・ユニットのデザインにおいて使用されている従来の演算要素をLPHDR演算要素で置き換えたり、補完したりすることができる。また、別の例では、本発明の実施例は、MIMDタイプのアーキテクチャーとして実行することもできる。そして、このアーキテクチャーには、ここで開示したいいずれかのタイプのLPHDR演算要素が取り入れられている。例えば、現在のあるいは新しいMIMDコンピュータ・システムのデザインにおいて使用されている従来の演算要素をLPHDR演算要素で置き換えたり、補完したりすることができる。更に、別の例では、本発明の実施例は、大規模並列処理マシンを含む、いずれかの種類のマシンとして実行することもできる。そしてこの大規模並列処理マシンは、従来のアーキテクチャーに比べ、少ないリソース(例えば、トランジスターや面積や体積など)を使用して大量の演算能力を提供するために、コンパクトな演算処理要素を使用している。

## 【0134】

ここでは、ソフトウェアを実行する、本発明にかかるある種の実施例について説明したが、これは単なる例示であって、本発明を限定するものではない。また、例えば、本発明の実施例は、ここで開示されたいいずれかの種類のLPHDR演算要素を制御するために、マイクロコードやハードウェア・シーケンサー、あるいはステート・マシンやその他のコントローラーを使用して実行することができる。また、例えば、本発明の実施例は、ここで開示されたいいずれかの種類のLPHDR演算要素を制御するために、ハード・ワイヤードの、焼き付けられた、あるいはプリ・プログラムド・コントローラーを使用して実行することができる。

## 【0135】

本発明のある種の実施例は、ハードウェアとしてカスタム・シリコンを使用して実行されるとして説明してきたが、これは単なる例示であって、本発明を限定するものではない。また、例えば、本発明の実施例は、FPGAや内在するハードウェアとしての、その他の再構成可能なチップを使用して実行することができる。そして、このFPGAsやその他の再構成可能なチップは、ここで開示したLPHDR処理を実行するために構成されている。別の例として、本発明の実施例は、プログラマブルで、従来のデジタルまたはアナログ計算のアーキテクチャー(高精度な計算要素を使用したものや、LPHDR演算を行うためのその他の種類の非LPHDRハードウェアを使用したものや、大規模並列処理するものなどが含まれている)を使用して実行することができる。そして、このアーキテクチャーは、ここで開示したLPHDR処理を実行するためのソフトウェアでプログラムされている。例えば、本発明の実施例は、ここで開示した機能のソフトウェア・エミュレータを使用して実行することができる。

## 【0136】

更に別の例として、本発明の実施例は、シリコン・チップ、あるいはその他の技術に基づくものであっても、3D組立技術を使用して実施することができる。いくつかの例示した実施例では、1つのメモリ・チップがプロセッサまたは別のデバイス上に接続されているか、あるいは数個のメモリ・チップ及び/又はプロセッサ・チップ、あるいはその他のデバイス・チップが互いにスタック状に重ねて接続されている。本発明の3D技術を採用した実施例は、2D技術を採用した実施例に比べ、高密度化されており、プロセッサ・ユニット間での3Dの情報通信が可能となるので、非常に有用である。そして、この3D技術を採用した実施例は、2D技術を採用した実施例に比べ、効率的に実行できる高度なアルゴリズムを可能ならしめるものである。

## 【0137】

本発明のある実施例では、シリコン・チップの組立技術を使って実施されるとして説明したが、これは単なる例示であって、本発明を限定するものではない。また、例えば、本発明の実施例は、その他の種類の従来のデジタルおよびアナログ形式のコンピュータ・プロセッサやその他のデバイスを使用する技術によって実行することができる。このような技術の例では、種々のナノメカニカル・テクノロジーやナノエレクトロニクス・テクノロジー、DNA計算のような化学ベースのテクノロジー、ナノワイヤーとナノチューブをベースとしたテクノロジー、オプティカル・テクノロジー、メカニカル・テクノロジー、バイオロジカル・テクノロジー、およびトランジスターに基礎を置くか否かは別として、その他の技術を含み、これらは、ここで開示した種類のLPHDR演算を実行できる可能性を持ったものである。

10

## 【0138】

本発明のある実施例では、「大規模並列」技術を使った実施例であるとして説明してきた。本発明のある実施例では、数千、数百万、あるいはそれ以上の演算ユニットを含んでいるが、本発明の実施例は、いくつかの(非常に少ない数の)演算ユニットを含むこともできる。例えば、単一のLPHDRユニットのみを含む実施例であっても、小さくて、安価なプロセッサやその他のデバイスにおいて大量のLPHDR処理できるパワーを提供するために、シリアル・プロセッシング・ユニットあるいはその他のデバイスの中で使用することができる。

## 【0139】

20

本発明のある実施例では、デジタル技術のみを使って実行されている場合であっても、演算処理は、選択した低精度の表示方式において、決定論的で、繰り返し可能で、あるいは最も正確である可能性のある結果を生み出すものではない。例えば、ある特定の入力値に対して、演算処理は、真の演算結果に対して、選択した表示方式において最も近い値ではない結果を生み出すかもしれない。

## 【0140】

「低精度のハイ・ダイナミック・レンジ」の演算要素の精度の程度は、個々の実施内容によって変化する。例えば、ある実施例では、LPHDR演算要素は小数、すなわち0より大きく1より小さい数を含む結果を生み出す。例えば、ある実施例では、LPHDR演算要素は、往々にして(あるいは常に)、正しい結果に対して0.05%以上離れた結果を生み出す。(すなわち、LPHDR演算要素によって生み出された結果と正しい結果の間の差の絶対値は、正しい結果の絶対値の1%の20分の1を超えないものである。)

30

別の実施例として、LPHDR演算要素は、往々にして(あるいは常に)、正しい結果に対して0.1%以上離れた結果を生み出す。

別の実施例として、LPHDR演算要素は、往々にして(あるいは常に)、正しい結果に対して0.2%以上離れた結果を生み出す。

更に、別の実施例として、LPHDR演算要素は、往々にして(あるいは常に)、正しい結果に対して0.5%以上離れた結果を生み出す。

更に、別の実施例として、LPHDR演算要素は、往々にして(あるいは常に)、正しい結果に対して1%、あるいは2%、あるいは5%、あるいは10%、あるいは20%以上離れた結果を生み出す。

40

## 【0141】

精度の程度については、いろいろな値をとる可能性があるが、この他に、LPHDR演算要素が処理する数値の空間のダイナミック・レンジによって、演算の実行は変化する。例えば、ある実施例では、LPHDR演算要素は、ほぼ100万分の1から100万までの範囲を有するレンジの空間において、数値を処理する。別の例として、ある実施例では、LPHDR演算要素は、ほぼ10億分の1から10億までの範囲を有するレンジの空間において、数値を処理する。

更に別の例として、ある実施例では、LPHDR演算要素は、ほぼ65,000分の1から65,000までの範囲を有するレンジの空間において、数値を処理する。更に別の例として、ある実施

50



(32)

JP 2012-530966 A 2012.12.6

例では、LPHDR演算要素は、0と65,000分の1の間にある特定の値から、65,000よりも大きい特定の値までの範囲を有するレンジの空間において、数値を処理する。更に別の例として、別の実施例であるLPHDR演算要素は、前述の例で示した範囲内あるいは前述の例で示した範囲を組み合わせたダイナミック・レンジ、例えば、ほぼ10億分の1から1000万までの範囲のレンジを備えた空間において、数値を処理することができる。別の実施例と同様、本発明として例示したこれらの全ての実施例において、議論の対象となっている数値は、符号の付いたものであり、そして、上述の議論は取り扱う数値の絶対値の特性を論ずるものである。

#### 【0142】

正しい結果に対する近似値のみを生み出すことができるLPHDR演算要素の周波数は、実行する演算の種類によって異なる。例えば、LPHDR演算要素が一つまたはそれ以上の処理(おそらく、例えば、三角関数を含むような処理)を実行することができるようになっていた実施例であって、各処理において、LPHDR演算要素の各々が有効な数値の範囲から取り出した入力値のセットを受け入れ、入力値の特定のセットに対しLPHDR演算要素が一つまたはそれ以上の出力値を生み出し(例えば、入力値のサインおよびコサインの両方を同時に計算するようなケース)、入力値の特定のセットに対して生み出された出力値が決定論的、あるいは非決定論的である場合について考えてみる。このような例の実施例では、有効な入力と相対的な誤差の大きさEの比Fについて更に考える。そして、このことによって、LPHDR演算要素によって計算される結果は、数学的に正しい結果とは異なったものとなる。本発明のある実施例では、各LPHDR演算要素の各々において、LPHDR演算ユニットが実行できる少なくとも1回の処理に関するものであって、当該処理に対する許容できる有効な入力の少なくとも比Fに関して、当該処理によって生み出される少なくとも一つの出力信号に関して、繰り返し実行される演算の全体にわたって求めた統計的平均であって、処理が個々の入力に対して実行された時の、LPHDR演算ユニットの出力信号によって表される数値の統計的平均は、同じ入力値に対する正しい数学的計算結果に比べて、少なくともEだけ異なる。ここで、Fは1%であり、Eは0.05%である。いくつかの別の実施例では、Fは1%ではなく、2%、または5%、または10%、または20%、または50%の内のいずれかである。これらの実施例の各々では、Fの値としていくつかの特定の値をとっており、Eの値として0.05%ではなく、0.1%、あるいは0.2%、あるいは0.5%、あるいは1%、あるいは2%、あるいは5%、あるいは10%、あるいは20%のE値を有するその他の実施例がある。これらの種々の実施例は、単に例示するものであり、本発明を限定するものではない。

#### 【0143】

本発明により具体化したあるデバイス(コンピュータや、プロセッサ、あるいはその他のデバイスのようなもの)では、デバイス内(例えば、コンピュータや、プロセッサ、あるいはその他のデバイス)のLPHDR演算要素の数は、従来の精度(すなわち、32ビットまたはそれ以上のビット数のワード長での浮動小数点演算)でハイ・ダイナミック・レンジの演算を実行するように設計されたデバイス内の演算要素の数(ゼロの場合もありうる)を超える。NLを、デバイス内のLPHDR演算要素の合計数とし、NHを、従来の精度でハイ・ダイナミック・レンジの演算を実行するように設計されたデバイス内の演算要素の合計数とすると、NLはT(NH)を超える。ここで、T( )は、ある関数を意味する。種々の関数の中のいずれかのものが関数T( )として使用することができる。例えば、ある実施例では、T(NH)は $20+3 \times NH$ であり、デバイス中のLPHDR演算要素の数は、もしデバイス内に演算要素があったとすれば、 $20+3 \times [\text{デバイス内の演算要素の数}]$ を超える。ここで、このデバイスは、従来の精度でハイ・ダイナミック・レンジの演算を実行するように設計されたものである。また、別の例として、ある実施例では、デバイス内のLPHDR演算要素の数は、もしデバイス内に演算要素があったとすれば、 $50+5 \times [\text{デバイス内の演算要素の数}]$ を超える。ここで、このデバイスは、従来の精度でハイ・ダイナミック・レンジの演算を実行するように設計されたものである。また、別の例として、ある実施例では、デバイス内のLPHDR演算要素の数は、もしデバイス内に演算要素があったとすれば、 $100+5 \times [\text{デバイス内の演算要素の数}]$ を超える。ここで、このデバイスは、従来の精度でハイ・ダ

(33)

JP 2012-530966 A 2012.12.6

イナミック・レンジの演算を実行するように設計されたものである。また、別の例として、ある実施例では、デバイス内のLPHDR演算要素の数は、もしデバイス内に演算要素があったとすれば、 $1000+5x[\text{デバイス内の演算要素の数}]$ を超える。ここで、このデバイスは、従来の精度でハイ・ダイナミック・レンジの演算を実行するように設計されたものである。また、別の例として、ある実施例では、デバイス内のLPHDR演算要素の数は、もしデバイス内に演算要素があったとすれば、 $5000+5x[\text{デバイス内の演算要素の数}]$ を超える。ここで、このデバイスは、従来の精度でハイ・ダイナミック・レンジの演算を実行するように設計されたものである。本発明のある実施例は、これに限定するものではないがシリコン・チップ、あるいはスタックされたチップ、あるいはチップ・パッケージ、あるいは回路ボードのような単一の物理デバイスの中で実行されるようになっている。

そして、物理デバイスの中のLPHDR演算要素の数NL、および物理デバイスの中で従来の精度でハイ・ダイナミック・レンジの演算を実行するように設計された演算要素の数NHは、その物理デバイスの中の個々の演算要素の合計数になる。本発明のある実施例は、一つ以上の物理的デバイスを含むコンピュータ・システムの中で実行される。そして、このような物理的デバイスには、これらに限定するものではないが、シリコン・チップ、あるいはスタックされたチップ、あるいはチップ・パッケージ、あるいは回路ボードの集合体であって、互いに結合され、種々の手段(バス、スイッチ、いずれかの種類のネットワーク接続、あるいはその他の通信手段)を使って互いに通信し合うようになっているものがある。そして、このケースでは、コンピュータ・システムの中のLPHDR演算要素の数NL、およびコンピュータ・システムの中で従来の精度でハイ・ダイナミック・レンジの演算を実行するように設計された演算要素の数NHは、これらの連結された全ての物理デバイスの中の個々の演算要素の合計数になる。

#### 【0144】

本発明のある実施例は、プロセッサあるいはプロセッサの一部を構成することができるようになる。このプロセッサは、計算を実行するためのソフトウェアを稼働する能力をもったデバイスである。かかるプロセッサは、ソフトウェアを保存するためのメカニズム、どの処理を実施すべきかを決定するためのソフトウェアを使うメカニズム、これらの処理を実行するためのメカニズム、大量のデータを保存するためのメカニズム、特定の処理を行うソフトウェアによってデータを修正するためのメカニズム、そしてプロセッサに接続されたデバイスと通信するためのメカニズム、を備えている。プロセッサは、特に限定するものではないが、フィールド・プログラマブル・アレイのような再構成可能なデバイスであっても良い。プロセッサは、ホスト・マシンをアシストするためのコプロセッサであっても良いし、外部のホスト・マシンから独立して演算処理できる能力を有するものであっても良い。プロセッサは、CPUs、GPUs、FPGAs、その他のプロセッサやその他のデバイスのような、種々のタイプのホスト・プロセッサおよびコプロセッサのコンポーネントの集合体として形成されるものであっても良い。そしてこれは、異機種環境のプロセッサ・デザイン、あるいは異種環境のコンピュータ・システムとして言及されるものであり、これらの内のいくつか、あるいはこれらの全てのコンポーネントは、本発明の実施例と同じか、そのバリエーションに含まれるものである。

#### 【0145】

しかし、本発明の実施例は、プロセッサに追加されたデバイス、あるいはプロセッサ以外のデバイスにおいて実行することができる。例えば、プロセッサとその他のコンポーネント(データ・バスによってプロセッサに接続されたメモリのようなもの)を含むコンピュータは、本発明の実施例の一例である。ここで、プロセッサは、本明細書で開示したいずれかの方法によりLPHDR処理を行うためのコンポーネントを含んでいる。

更に一般的には、プロセッサという意味の範囲に入るか否かは別として、ここで開示した機能を実行するいずれかのデバイスまたはデバイスの組み合わせは、本発明の実施例に関する一例を構成するものである。

#### 【0146】

(34)

JP 2012-530966 A 2012.12.6

更に一般的には、ここで開示した技術のいずれかは、例えば、ハードウェア、コンピュータ読み取り可能な媒体に確実に記憶されたソフトウェア、ファームウェア、あるいはこれらの組み合わせによって実行される。ここで述べた技術は、プロセッサ、プロセッサによる読み取り可能な記憶媒体(例えば、揮発性あるいは非揮発性のメモリおよび/または記憶要素を含む)、少なくとも1つの入力デバイス、および少なくとも一つの出力デバイスを備えたプログラマブル・コンピュータ上で実行される一つまたはそれ以上のコンピュータ・プログラムによって実行することができる。プログラム・コードは、上述した機能を実行し出力を生み出すために、入力デバイスを使って、入力される。出力は、一つまたはそれ以上の出力デバイスに供給される。

【0147】

10

後で述べるクレームの範囲内となる各コンピュータ・プログラムは、アセンブリ言語、機械言語、ハイレベルなプロシージャル・プログラミング言語、あるいは、オブジェクト・オリエンテッド言語のように、どのようなプログラミング言語であっても実行することができる。例えば、プログラミング言語はコンパイルされ、解釈される。

【0148】

かかるコンピュータ・プログラムの各々は、コンピュータ・プロセッサによって実行するために、マシンによる読み取り可能な記憶デバイスの中に確実に具体化されたコンピュータ・プログラム製品の形態で実行される。この発明の手法は、入力を処理し出力を生み出すことによって発明の機能を実行させるために、コンピュータによる読み取り可能な記憶媒体の中に確実に具体化されたプログラムを稼働させるコンピュータ・プロセッサによって実行される。適切なプロセッサには、例えば、汎用目的および特殊用途目的の両方のマイクロプロセッサが含まれる。一般的に、このプロセッサは、命令とデータをリード・オンリー・メモリおよび/またはランダム・アクセス・メモリから受けとる。コンピュータ・プログラム命令を確実に具体化するのに適した記憶デバイスには、例えば、EPROMやEEPROMを含む半導体メモリ・デバイス、フラッシュ・メモリ・デバイス、内部ハードディスクやリムーバブル・ディスクなどの磁気ディスク、光-磁気ディスク、及びCR-ROMs等の全てのタイプの非揮発性のメモリがある。ここで述べたもののいずれもが、特別に設計されたASICs(Application-Specific Integrated Circuits)やFPGAs(Field-Programmable Gate Arrays)によって取り込まれ、実行することができるものである。また、コンピュータは、一般に、内部ハード・ディスク(図示せず)あるいはリムーバブル・ディスクのような記憶媒体からプログラムとデータを受け取る。これらの要素は、従来のデスクトップやワークステーションコンピュータ、そしてここで述べた方法を実行するコンピュータ・プログラムを駆動するのに適したその他のコンピュータにおいても見いだされるものである。そしてこれらの要素は、デジタル・プリント・エンジンやマーキング・エンジン、ディスプレイ・モニター、および、紙、フィルム、ディスプレイ・スクリーンあるいはその他の出力媒体の上にカラーやグレイ・スケールのピクセルを出力する機能を持った、その他のラスター・アウトプット・デバイスと共に使用されるものである。

20

30

40

(35)

JP 2012-530966 A 2012.12.6

【図 1】

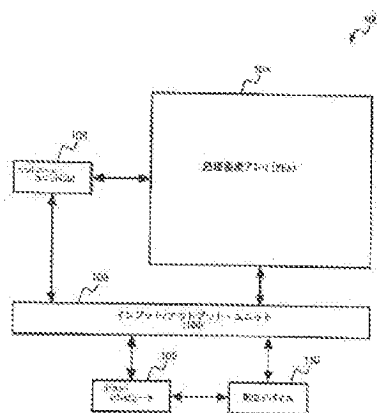


FIG. 1

【図 2】

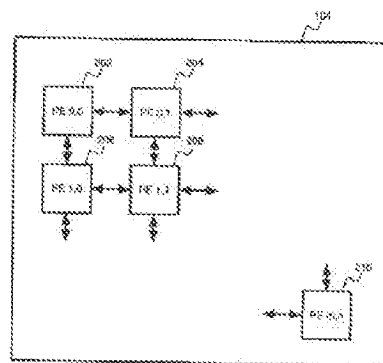


FIG. 2

【図 3】

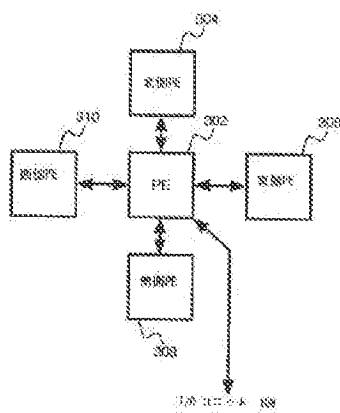


FIG. 3

【図 4】

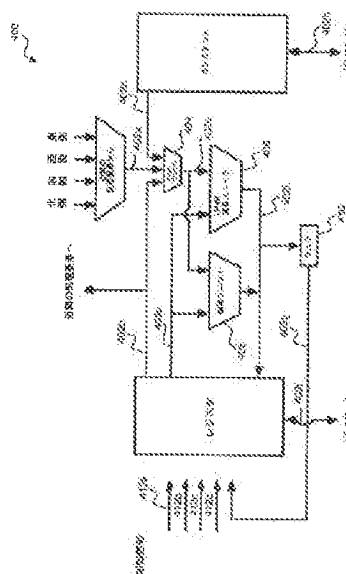


FIG. 4

(36)

JP 2012-530966 A 2012.12.6

【図 5】

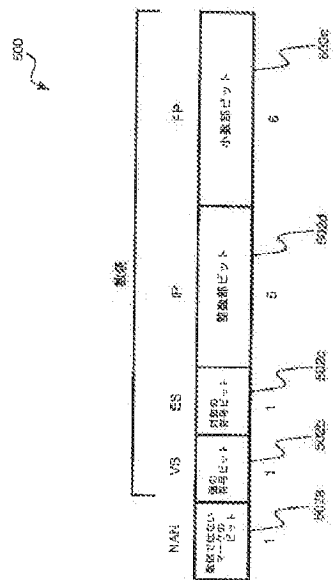


FIG. 5

【図 6】

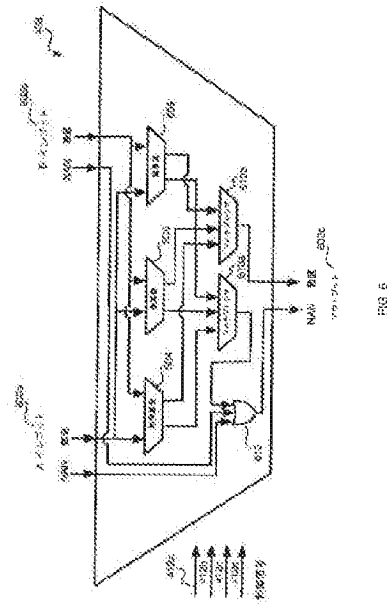


FIG. 6

【図 7】

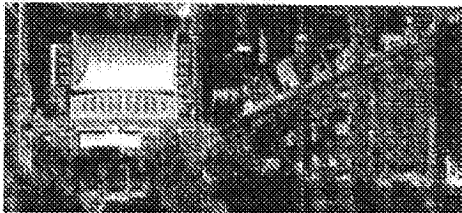


FIG. 7

【図 9】



FIG. 9

【図 8】

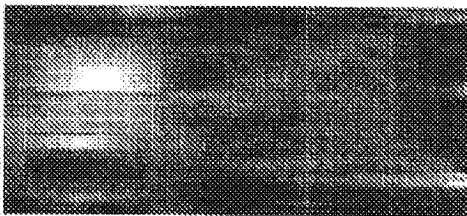


FIG. 8

【図 10】

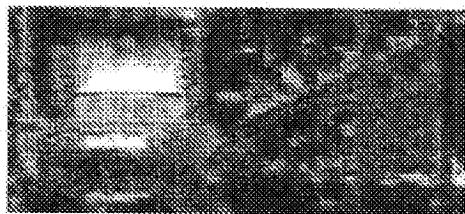


FIG. 10



(37)

JP 2012-530966 A 2012.12.6

【図 11】

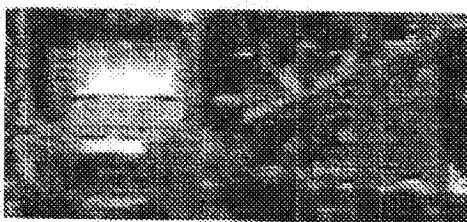


FIG. 11

【手続補正書】

【提出日】平成23年4月28日(2011.4.28)

【手続補正1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】

少なくとも1つの第1の低精度ハイ・ダイナミック・レンジ(LPHDR)演算実行ユニットであって、第2の数値を表す第1の出力信号を生み出すために、第1の数値を表す第1の入力信号に対して第1の処理を実行するように構成されており、

第1の処理への許容できる有効な入力ダイナミック・レンジは、少なくとも1/65,000から65,000までの幅を有し、第1の処理への許容できる有効な入力の少なくとも $X=5\%$ に關して、第1の処理への許容できる有効な入力の少なくとも $X\%$ から各特定の入力までの繰り返し実行される第1の処理において、その入力に対する第1の処理を実行するLPHDR演算実行ユニットの第1の出力信号によって表される数値の統計学的平均が、その同じ入力の数値に対する第1の処理の正確な数学的計算の結果より少なくとも $Y=0.05\%$ だけ異なるLPHDR演算実行ユニットを備えることを特徴とするデバイス。

【請求項2】

請求項1に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、フィールド・プログラマブル・ゲート・アレイ(FPGA)の少なくとも一部を備えていることを特徴とするデバイス。

【請求項3】

請求項1に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニ



(38)

JP 2012-530966 A 2012.12.6

ットは、少なくとも10個のLPHDR演算実行ユニットを備えていることを特徴とするデバイス。

【請求項4】

請求項3に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、フィールド・プログラマブル・ゲート・アレイ(FPGA)の少なくとも一部を備えていることを特徴とするデバイス。

【請求項5】

請求項1に記載されたデバイスであって、前記デバイス中のLPHDR演算実行ユニットの数は、 $N$ 、  
少なくとも32ビット幅の浮動小数点数に対する少なくとも乗算の演算を実行するように構成されたデバイスにおける、負ではない整数の演算実行ユニットの数に比べて、  
少なくとも10多くなっていることを特徴とするデバイス。

【請求項6】

請求項5に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、フィールド・プログラマブル・ゲート・アレイ(FPGA)の少なくとも一部を備えていることを特徴とするデバイス。

【請求項7】

請求項1に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、少なくとも100個のLPHDR演算実行ユニットを備えていることを特徴とするデバイス。

【請求項8】

請求項1に記載されたデバイスであって、前記デバイス中のLPHDR演算実行ユニットの数は、 $N$ 、  
少なくとも32ビット幅の浮動小数点数に対する少なくとも乗算の演算を実行するように構成されたデバイスにおける、負ではない整数の演算実行ユニットの数に比べて、  
少なくとも100多くなっていることを特徴とするデバイス。

【請求項9】

請求項1に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、少なくとも500個のLPHDR演算実行ユニットを備えていることを特徴とするデバイス。

【請求項10】

請求項1に記載されたデバイスであって、前記デバイス中のLPHDR演算実行ユニットの数は、 $N$ 、  
少なくとも32ビット幅の浮動小数点数に対する少なくとも乗算の演算を実行するように構成されたデバイスにおける、負ではない整数の演算実行ユニットの数に比べて、  
少なくとも500多くなっていることを特徴とするデバイス。

【請求項11】

コンピュータ・プロセッサと、コンピュータ・プログラムの命令を記憶するためのコンピュータ読み取り可能なメモリを備え、

当該コンピュータ・プログラムの命令は、第2のデバイスをエミュレートするためのプロセッサによって実行することができ、

当該第2のデバイスは、

少なくとも1つの第1の低精度ハイ・ダイナミック・レンジ(LPHDR)演算実行ユニットであって、第2の数値を表す第1の出力信号を生み出すために、第1の数値を表す第1の入力信号に対して第1の処理を実行するように構成されており、

第1の処理への許容できる有効な入力ダイナミック・レンジは、少なくとも1/65,000から65,000までの幅を有し、第1の処理への許容できる有効な入力の少なくとも $X=5\%$ に関して、第1の処理への許容できる有効な入力の少なくとも $X\%$ から各特定の入力までの繰返し実行される第1の処理において、その入力に対する第1の処理を実行するLPHDR演算実行ユニットの第1の出力信号によって表される数値の統計学的平均が、その同じ入力の数

(39)

JP 2012-530966 A 2012.12.6

値に対する第1の処理の正確な数学的計算の結果より少なくとも $Y=0.05\%$ だけ異なる LPHDR演算実行ユニットを備えることを特徴とするデバイス。

【請求項12】

請求項11に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、少なくとも10個のLPHDR演算実行ユニットを備えていることを特徴とするデバイス。

【請求項13】

請求項11に記載されたデバイスであって、前記第2のデバイス中のLPHDR演算実行ユニットの数は、  
少なくとも32ビット幅の浮動小数点数に対する少なくとも乗算の演算を実行するように構成された第2のデバイスにおける、負ではない整数の演算実行ユニットの数に比べて、少なくとも10多くなっていることを特徴とするデバイス。

【請求項14】

少なくとも1つの第1の低精度ハイ・ダイナミック・レンジ(LPHDR)演算実行ユニットであって、第2の数値を表す第1の出力信号を生み出すために、第1の数値を表す第1の入力信号に対して第1の処理を実行するように構成されており、

第1の処理への許容できる有効な入力ダイナミック・レンジは、少なくとも1/65,000から65,000までの幅を有し、第1の処理への許容できる有効な入力の少なくとも $X=5\%$ に関して、第1の処理への許容できる有効な入力の少なくとも $X\%$ から各特定の入力までの繰り返し実行される第1の処理において、その入力に対する第1の処理を実行するLPHDR演算実行ユニットの第1の出力信号によって表される数値の統計学的平均が、その同じ入力の数値に対する第1の処理の正確な数学的計算の結果より少なくとも $Y=0.05\%$ だけ異なる LPHDR演算実行ユニットを備え、

当該デバイス中のLPHDR演算実行ユニットの数は、

少なくとも32ビット幅の浮動小数点数に対する少なくとも乗算の演算を実行するように構成されたデバイスにおける、負ではない整数の演算実行ユニットの数を上回っていることを特徴とするデバイス。

【請求項15】

請求項14に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、フィールド・プログラマブル・ゲート・アレイ(FPGA)の少なくとも一部を備えていることを特徴とするデバイス。

【請求項16】

請求項14に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、少なくとも10個のLPHDR演算実行ユニットを備えていることを特徴とするデバイス。

【請求項17】

請求項16に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、フィールド・プログラマブル・ゲート・アレイ(FPGA)の少なくとも一部を備えていることを特徴とするデバイス。

【請求項18】

請求項14に記載されたデバイスであって、前記デバイス中のLPHDR演算実行ユニットの数は、

少なくとも32ビット幅の浮動小数点数に対する少なくとも乗算の演算を実行するように構成されたデバイスにおける、負ではない整数の演算実行ユニットの数に比べて、少なくとも10多くなっていることを特徴とするデバイス。

【請求項19】

請求項18に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、フィールド・プログラマブル・ゲート・アレイ(FPGA)の少なくとも一部を備えていることを特徴とするデバイス。

【請求項20】

(40)

JP 2012-530966 A 2012.12.6

請求項14に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、少なくとも100個のLPHDR演算実行ユニットを備えていることを特徴とするデバイス。

【請求項21】

請求項14に記載されたデバイスであって、前記デバイス中のLPHDR演算実行ユニットの数は、少なくとも32ビット幅の浮動小数点数に対する少なくとも乗算の演算を実行するように構成されたデバイスにおける、負ではない整数の演算実行ユニットの数に比べて、少なくとも100多くなっていることを特徴とするデバイス。

【請求項22】

請求項14に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、少なくとも500個のLPHDR演算実行ユニットを備えていることを特徴とするデバイス。

【請求項23】

請求項14に記載されたデバイスであって、前記デバイス中のLPHDR演算実行ユニットの数は、少なくとも32ビット幅の浮動小数点数に対する少なくとも乗算の演算を実行するように構成されたデバイスにおける、負ではない整数の演算実行ユニットの数に比べて、少なくとも500多くなっていることを特徴とするデバイス。

【請求項24】

コンピュータ・プロセッサと、コンピュータ・プログラムの命令を記憶するためのコンピュータ読み取り可能なメモリを備え、

当該コンピュータ・プログラムの命令は、第2のデバイスをエミュレートするためのプロセッサによって実行することができ、

当該第2のデバイスは、

少なくとも1つの第1の低精度ハイ・ダイナミック・レンジ(LPHDR)演算実行ユニットであって、第2の数値を表す第1の出力信号を生み出すために、第1の数値を表す第1の入力信号に対して第1の処理を実行するように構成されており、

第1の処理への許容できる有効な入力のダイナミック・レンジは、少なくとも1/65,000から65,000までの幅を有し、第1の処理への許容できる有効な入力の少なくとも5%に關して、第1の処理への許容できる有効な入力の少なくとも5%から各特定の入力までの繰り返し実行される第1の処理において、その入力に対する第1の処理を実行するLPHDR演算実行ユニットの第1の出力信号によって表される数値の統計学的平均が、その同じ入力の数値に対する第1の処理の正確な数学的計算の結果より少なくとも0.05%だけ異なるLPHDR演算実行ユニットを備え、

当該第2のデバイス中のLPHDR演算実行ユニットの数は、

少なくとも32ビット幅の浮動小数点数に対する少なくとも乗算の演算を実行するように構成された第2のデバイスにおける、負ではない整数の演算実行ユニットの数を上回っていることを特徴とするデバイス。

【請求項25】

請求項24に記載されたデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットは、少なくとも10個のLPHDR演算実行ユニットを備えていることを特徴とするデバイス。

【請求項26】

請求項24に記載されたデバイスであって、前記第2のデバイス中のLPHDR演算実行ユニットの数は、

少なくとも32ビット幅の浮動小数点数に対する少なくとも乗算の演算を実行するように構成された第2のデバイスにおける、負ではない整数の演算実行ユニットの数に比べて、少なくとも10多くなっていることを特徴とするデバイス。

【請求項27】

(41)

JP 2012-530966 A 2012.12.6

請求項1乃至26のいずれかに記載されたデバイスであって、前記Xが $X=10\%$ であることを特徴とするデバイス。

【請求項28】

請求項1乃至26のいずれかに記載されたデバイスであって、前記Yが $Y=0.1\%$ であることを特徴とするデバイス。

【請求項29】

請求項1乃至26のいずれかに記載されたデバイスであって、前記Yが $Y=0.15\%$ であることを特徴とするデバイス。

【請求項30】

請求項1乃至26のいずれかに記載されたデバイスであって、前記Yが $Y=0.2\%$ であることを特徴とするデバイス。

【請求項31】

請求項1乃至26のいずれかに記載されたデバイスであって、前記Xが $X=10\%$ であり、前記Yが $Y=0.1\%$ であることを特徴とするデバイス。

【請求項32】

請求項1乃至26のいずれかに記載されたデバイスであって、前記Xが $X=10\%$ であり、前記Yが $Y=0.15\%$ であることを特徴とするデバイス。

【請求項33】

請求項1乃至26のいずれかに記載されたデバイスであって、前記Xが $X=10\%$ であり、前記Yが $Y=0.2\%$ であることを特徴とするデバイス。

【請求項34】

請求項1乃至26のいずれかに記載されたデバイスであって、第1の処理への入力であって、許容できる有効な入力のダイナミック・レンジは、少なくとも $1/1,000,000$ から $1,000,000$ までの幅を有することを特徴とするデバイス。

【請求項35】

請求項1、11、14、24のいずれかに記載のデバイスであって、少なくとも1つの第1のLPHDR演算実行ユニットが、複数の局所的に接続されたLPHDR演算実行ユニットからなることを特徴とするデバイス。

【請求項36】

請求項1、11、14、24のいずれかに記載のデバイスであって、前記デバイスがSIMDアーキテクチャを備えていることを特徴とするデバイス。

【請求項37】

請求項1、11、14、24のいずれかに記載のデバイスであって、前記デバイスは、少なくとも1つの第1のLPHDR演算実行ユニットに局所的にアクセス可能なメモリを備えていることを特徴とするデバイス。

【請求項38】

請求項1、11、14、24のいずれかに記載のデバイスであって、前記デバイスは、シリコン・チップ上に実装されていることを特徴とするデバイス。

【請求項39】

請求項1、11、14、24のいずれかに記載のデバイスであって、前記デバイスは、デジタル技術を使用し、シリコン・チップ上に実装されていることを特徴とするデバイス。

【請求項40】

請求項1、11、14、24のいずれかに記載のデバイスであって、前記デバイスは、少なくとも1つの第1のLPHDR演算実行ユニットを制御するように構成されたデジタル・プロセッサを更に備えている。

【請求項41】

請求項1、11、14、24のいずれかに記載のデバイスであって、前記少なくとも1つのLPHDR演算実行ユニットは、少なくとも500個の局所的に接続されたLPHDR演算実行ユニットからなり、前記デバイスは、少なくとも1つのLPHDR演算実行ユニットに局所的にアクセス可能なメモ



(42)

JP 2012-530966 A 2012.12.6

リを備えており、

前記デバイスは、デジタル技術を使用し、シリコン・チップ上に実装されていることを特徴とするデバイス。

【請求項42】

請求項1、11、14、24のいずれかに記載のデバイスであって、前記デバイスはモバイル・デバイスの一部であることを特徴とするデバイス。

【請求項43】

請求項1、11、14、24のいずれかに記載のデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットが、対数表示を使用することによって数値を表わしていることを特徴とするデバイス。

【請求項44】

請求項1、11、14、24のいずれかに記載のデバイスであって、前記少なくとも1つの第1のLPHDR演算実行ユニットが、浮動小数点表示を使用することによって数値を表わしていることを特徴とするデバイス。

【請求項45】

請求項1、11、14、24のいずれかに記載のデバイスであって、前記デバイスは、入力画像を表すデータを受け取るための入力手段を更に備え、

当該入力画像には第1の入力信号が含まれていることを特徴とするデバイス。

【請求項46】

請求項45に記載のデバイスであって、前記デバイスはモバイル・デバイスの一部であることを特徴とするデバイス。

【請求項47】

請求項45に記載のデバイスであって、前記デバイスは入力画像のぼやけ除去のために構成されたものであることを特徴とするデバイス。

【請求項48】

請求項1、11、14、24のいずれかに記載のデバイスであって、前記デバイスは、最近傍探索のために構成されたものであることを特徴とするデバイス。

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0140

【補正方法】変更

【補正の内容】

【0140】

「低精度のハイ・ダイナミック・レンジ」の演算要素の精度の程度は、個々の実施内容によって変化する。例えば、ある実施例では、LPHDR演算要素は小数、すなわち0より大きく1より小さい数を含む結果を生み出す。例えば、ある実施例では、LPHDR演算要素は、往々にして(あるいは常に)、正しい結果に対して0.05%以上離れた結果を生み出す。(すなわち、LPHDR演算要素によって生み出された結果と正しい結果の間の差の絶対値は、正しい結果の絶対値の1%の20分の1を超えるものである。)

別の実施例として、LPHDR演算要素は、往々にして(あるいは常に)、正しい結果に対して0.1%以上離れた結果を生み出す。

別の実施例として、LPHDR演算要素は、往々にして(あるいは常に)、正しい結果に対して0.2%以上離れた結果を生み出す。


更に、別の実施例として、LPHDR演算要素は、往々にして(あるいは常に)、正しい結果に対して0.5%以上離れた結果を生み出す。

更に、別の実施例として、LPHDR演算要素は、往々にして(あるいは常に)、正しい結果に対して1%、あるいは2%、あるいは5%、あるいは10%、あるいは20%以上離れた結果を生み出す。

(43)

JP 2012-530966 A 2012.12.6

【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US2010/038769
<b>A. CLASSIFICATION OF SUBJECT MATTER</b>		
G06F 9/38(2006.01); G06F 9/46(2006.01); G06F 13/14(2006.01)		
According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b>		
Minimum documentation searched (classification system followed by classification symbols) G06F 9/38; G06F 9/40; G06F 9/302; G06F 9/28; G06F 15/167		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords: floating point execution unit, integer execution unit, multiprocessor		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No
A	US 5287160 A1 (LAURITZEN, M. et al.) 28 March 1999 See abstract, column 1 line 65 - column 2 line 64, column 3 line 15 - line 48, and figure 1.	1,2
A	US 5867683 A1 (WITT, D.B. et al.) 02 February 1999 See abstract and column 9 line 25 - line 35	1,2
A	US 5809320 A1 (JAIN, A. et al.) 15 September 1998 See abstract and claim 4.	1,2
A	US 5293500 A1 (ISHIDA, R. et al.) 08 March 1994 See abstract, column 1 line 65 - column 2 line 23, column 7 line 23 - column 8 line 17, and figure 9.	1,2
A	US 5235165 A1 (ISHIDA, R. et al.) 06 July 1993 See abstract, column 1 line 55 - column 2 line 14, column 2, line 36 - column 3 line 32, and figure 1.	1,2
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "I" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 27 JANUARY 2011 (27.01.2011)		Date of mailing of the international search report 28 JANUARY 2011 (28.01.2011)
Name and mailing address of the ISA/EK  Korean Intellectual Property Office Government Complex-Daejeon, 139 Seosun-ro, Seo-gu, Daejeon 302-701, Republic of Korea Facsimile No. 42-42-472-7190		Authorized officer Hwang, Seung Hee Telephone No.

Form PCT/ISA/210 (second sheet) (July 2009)



(44)

JP 2012-530966 A 2012.12.6

**INTERNATIONAL SEARCH REPORT**  
Information on patent family members

International application No.  
**PCT/US2010/038769**

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5887160 A1	23.03.1999	JP 03-607476 B2 JP 10-228378 A KR 10-0507415 B1	05.01.2005 25.08.1998 06.12.2005
US 5867683 A1	02.02.1999	EP 0651321 A1 EP 0651321 B1 EP 0679992 A1 EP 0679992 B1 JP 03-618621 B2 JP 03-670039 B2 JP 07-182180 A JP 07-295813 A US 05867683A A US 5574928 A1 US 5681125 A1 US 5655097 A1 US 5655088 A1 US 5664136 A1 US 5751981 A1 US 5867682 A1 US 5903772 A1	03.05.1995 14.11.2001 02.11.1995 23.01.2002 09.02.2005 13.07.2005 21.07.1995 10.11.1995 02.02.1999 12.11.1996 22.07.1997 05.08.1997 05.08.1997 02.09.1997 12.05.1998 02.02.1999 11.05.1999
US 5899320 A1	15.09.1998	EP 0463966 A2 EP 0463966 A3 EP 0463966 B1 JP 02-939003 B2 JP 06-075747 A JP 2939003 B2	02.01.1992 30.08.1993 25.11.1998 11.06.1999 18.03.1994 25.03.1998
US 5293500 A1	08.03.1994	JP 02-301829 A JP 02-301830 A JP 03-001232 A JP 07-089825 B2 JP 2058240 C JP 2211534 A US 5226166 A1	13.12.1990 13.12.1990 07.01.1991 31.07.1995 10.06.1996 22.08.1990 06.07.1993
US 5226166 A1	06.07.1993	JP 02-301829 A JP 02-301830 A JP 03-001232 A JP 07-089825 B2 JP 2058240 C JP 2211534 A US 05226166A A US 5293500 A1	13.12.1990 13.12.1990 07.01.1991 31.07.1995 10.06.1996 22.08.1990 06.07.1993 06.03.1994

Form PCT/ISA/210 (patent family annex) (July 2009)

(45)

JP 2012-530966 A 2012.12.6

フロントページの続き

(01)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LJ, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GR, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW



Espacenet

**Bibliographic data: JP2012530966 (A) — 2012-12-06**

**Processing with Compact Arithmetic Processing Element**

**Inventor(s):**

**Applicant(s):**

**Classification:** - international: G06F7/00; G06F9/38  
- cooperative: G06F7/38; G06F7/4833; G06F7/5235;  
H03K19/17728

**Application number:** JP20120516227 20100616

**Priority number (s):** US20090218691P 20090619 ; US20100816201 20100615 ;  
WO2010US38769 20100616

**Also published as:** US2010325186 (A1) US8150902 (B2) US2014095571 (A1)  
US2013031153 (A1) US2013031153 (A1) more

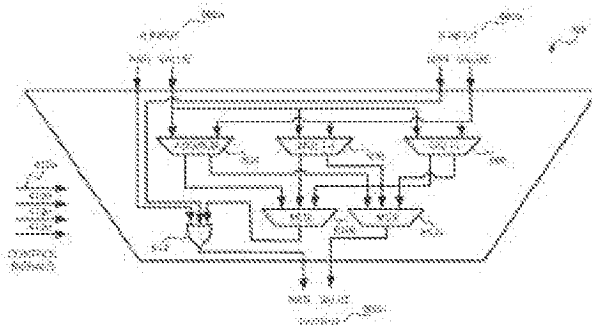
**Abstract not available for JP2012530966 (A)**

**Abstract of corresponding document: US2010325186 (A1)**

A processor or other device, such as a programmable and/or massively parallel processor or other device, includes processing elements designed to perform arithmetic operations (possibly but not necessarily including, for example, one or more of addition, multiplication,

subtraction, and division) on numerical values of low precision but high dynamic

range ("LPHDR arithmetic"). Such a processor or other device may, for example, be implemented on a single chip.; Whether or not implemented on a single chip, the number of LPHDR arithmetic elements in the processor or other device in certain embodiments of the present invention significantly exceeds (e.g., by at least 20 more than three times) the number of arithmetic elements, if any, in the processor or other device which are designed to perform high dynamic range arithmetic of traditional precision (such as 32 bit or 64 bit floating point arithmetic).





Espacenet

**Description: JP2012530966 (A) — 2012-12-06**

---

**Processing with Compact Arithmetic Processing Element**

**Description not available for JP2012530966 (A)**

**Description of corresponding document: US2010325186 (A1)**

**A high quality text as facsimile in your desired language may be available amongst the following family members:**

CA2768731 (A1) KR20120040197 (A) US2010325186 (A1) WO2010148054 (A2)  
US2013031153 (A1) US2014095571 (A1)

The EPO does not accept any responsibility for the accuracy of data and information originating from other authorities than the EPO; in particular, the EPO does not guarantee that they are complete, up-to-date or fit for specific purposes.

#### CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of commonly-owned and co-pending U.S. Provisional Patent Application Ser. No. 61/218,691, filed on Jun. 19, 2009, entitled, "Massively Parallel Processing with Compact Arithmetic Element," which is hereby incorporated by reference herein.

#### COPYRIGHT NOTICE

[0002] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

#### BACKGROUND

[0003] The ability to compute rapidly has become enormously important to humanity. Weather and climate prediction, medical applications (such as drug design and non-invasive imaging), national defense, geological exploration, financial modeling, Internet search, network communications, scientific research in varied fields, and even the design of new computing hardware have each become dependent on the ability to rapidly perform massive amounts of calculation. Future progress, such as the computer-aided design of complex nano-scale systems or development of consumer

products that can see, hear, and understand, will demand economical delivery of even greater computing power.

[0004] Gordon Moore's prediction, that computing performance per dollar would double every two years, has proved valid for over 30 years and looks likely to continue in some form. But despite this rapid exponential improvement, the reality is that the inherent computing power available from silicon has grown far more quickly than it has been made available to software. In other words, although the theoretical computing power of computing hardware has grown exponentially, the interfaces through which software is required to access the hardware limits the ability of software to use hardware to perform computations at anything approaching the hardware's theoretical maximum computing power.

[0005] Consider a modern silicon microprocessor chip containing about one billion transistors, clocked at roughly 1 GHz. On each cycle the chip delivers approximately one useful arithmetic operation to the software it is running. For instance, a value might be transferred between registers, another value might be incremented, perhaps a multiply is accomplished. This is not terribly different from what chips did 30 years ago, though the clock rates are perhaps a thousand times faster today.

[0006] Real computers are built as physical devices, and the underlying physics from which the machines are built often exhibits complex and interesting behavior. For example, a silicon MOSFET transistor is a device capable of performing interesting non-linear operations, such as exponentiation. The junction of two wires can add currents. If configured properly, a billion transistors and wires should be able to perform some significant fraction of a billion interesting computational operations within a few propagation delays of the basic components (a "cycle" if the overall design is a traditional digital design). Yet, today's CPU chips use their billion transistors to enable software to perform merely a few such operations per cycle, not the significant fraction of the billion that might be possible.

## SUMMARY

[0007] Embodiments of the present invention are directed to a processor or other device, such as a programmable and/or massively parallel processor or other device, which includes processing elements designed to perform arithmetic operations (possibly but not necessarily including, for example, one or more of addition, multiplication, subtraction, and division) on numerical values of low precision but high dynamic range ("LPHDR arithmetic"). Such a processor or other device may, for example, be implemented on a single chip. Whether or not implemented on a single chip, the number of LPHDR arithmetic elements in the processor or other device in certain embodiments of the present invention significantly exceeds (e.g., by at least 20 more than three times) the number of arithmetic elements in the processor or other device which are designed to perform high dynamic range arithmetic of traditional precision (such as 32 bit or 64 bit floating point arithmetic).

[0008] In some embodiments, "low precision" processing elements perform arithmetic operations which produce results that frequently differ from exact results by at least 0.1% (one tenth of one percent). This is far worse precision than the widely used IEEE 754 single precision floating point standard. Programmable embodiments of the present invention may be programmed with algorithms that function adequately despite these unusually large relative errors. In some embodiments, the processing elements



have "high dynamic range" in the sense that they are capable of operating on inputs and/or producing outputs spanning a range at least as large as from one millionth to one million.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1 is an example overall design of a SIMD processor according to one embodiment of the present invention.

[0010] FIG. 2 is an example of the Processing Element Array of a SIMD processor according to one embodiment of the present invention.

[0011] FIG. 3 is an example of how a Processing Element in a Processing Element Array communicates data with other parts of the processor according to one embodiment of the present invention.

[0012] FIG. 4 is an example design for a Processing Element according to one embodiment of the present invention.

[0013] FIG. 5 is an example LPHDR data word format according to one embodiment of the present invention.

[0014] FIG. 6 is an example design for an LPHDR arithmetic unit according to one embodiment of the present invention.

[0015] FIG. 7 is an original image.

[0016] FIG. 8 is an image blurred by a blur kernel according to one embodiment of the present invention.

[0017] FIG. 9 is an image produced by Richardson Lucy deconvolution using floating point arithmetic according to one embodiment of the present invention.

[0018] FIG. 10 is an image produced by Richardson Lucy deconvolution using LPHDR floating point arithmetic with added noise (fp+noise) according to one embodiment of the present invention.

[0019] FIG. 11 is an image produced by Richardson Lucy deconvolution using LPHDR logarithmic arithmetic (lns) according to one embodiment of the present invention.

#### DETAILED DESCRIPTION

[0020] As described above, today's CPU chips make inefficient use of their transistors. For example, a conventional CPU chip containing a billion transistors might enable software to perform merely a few operations per clock cycle. Although this is highly inefficient, those having ordinary skill in the art design CPUs in this way for what are widely accepted to be valid reasons. For example, such designs satisfy the (often essential) requirement for software compatibility with earlier designs. Furthermore, they deliver great precision, performing exact arithmetic with integers typically 32 or 64 bits long and performing rather accurate and widely standardized arithmetic with 32 and 64 bit floating point numbers. Many applications need this kind of precision. As a result, conventional CPUs typically are designed to provide such precision, using on the order



of a million transistors to implement the arithmetic operations.

[0021] There are many economically important applications, however, which are not especially sensitive to precision and that would greatly benefit, in the form of application performance per transistor, from the ability to draw upon a far greater fraction of the computing power inherent in those million transistors. Current architectures for general purpose computing fail to deliver this power.

[0022] Because of the weaknesses of conventional computers, such as typical microprocessors, other kinds of computers have been developed to attain higher performance. These machines include single instruction stream/multiple data stream (SIMD) designs, multiple instruction stream/multiple data stream (MIMD) designs, reconfigurable architectures such as field programmable gate arrays (FPGAs), and graphics processing unit designs (GPUs) which, when applied to general purpose computing, may be viewed as single instruction stream/multiple thread (SIMT) designs.

[0023] SIMD machines follow a sequential program, with each instruction performing operations on a collection of data. They come in two main varieties: vector processors and array processors. Vector processors stream data through a processing element (or small collection of such elements). Each component of the data stream is processed similarly. Vector machines gain speed by eliminating many instruction fetch/decode operations and by pipelining the processor so that the clock speed of the operations is increased.

[0024] Array processors distribute data across a grid of processing elements (PEs). Each element has its own memory. Instructions are broadcast to the PEs from a central control until, sequentially. Each PE performs the broadcast instruction on its local data (often with the option to sit idle that cycle). Array processors gain speed by using silicon efficiently—using just one instruction fetch/decode unit to drive many small simple execution units in parallel.

[0025] Array processors have been built using fixed point arithmetic at a wide variety of bit widths, such as 1, 4, 8, and wider, and using floating point arithmetic. Small bit widths allow the processing elements to be small, which allows more of them to fit in the computer, but many operations must be carried out in sequence to perform conventional arithmetic calculations. Wider widths allow conventional arithmetic operations to be completed in a single cycle. In practice, wider widths are desirable. Machines that were originally designed with small bit widths, such as the Connection Machine-1 and the Goodyear Massively Parallel Processor, which each used 1 bit wide processing elements, evolved toward wider data paths to better support fast arithmetic, producing machines such as the Connection Machine-2 which included 32 bit floating point hardware and the MasPar machines which succeeded the Goodyear machine and provided 4 bit processing elements in the MasPar-1 and 32 bit processing elements in the MasPar-2.

[0026] Array processors also have been designed to use analog representations of numbers and analog circuits to perform computations. The SCAMP is such a machine. These machines provide low precision arithmetic, in which each operation might introduce perhaps an error of a few percentage points in its results. They also introduce noise into their computations, so the computations are not repeatable. Further, they represent only a small range of values, corresponding for instance to 8 bit fixed point values rather than providing the large dynamic range of typical 32 or 64 bit floating

point representations. Given these limitations, the SCAMP was not intended as a general purpose computer, but instead was designed and used for image processing and for modeling biological early vision processes. Such applications do not require a full range of arithmetic operations in hardware, and the SCAMP, for example, omits general division and multiplication from its design.

[0027] While SIMD machines were popular in the 1980s, as price/performance for microprocessors improved designers began building machines from large collections of communicating microprocessors. These MIMD machines are fast and can have price/performance comparable to their component microprocessors, but they exhibit the same inefficiency as those components in that they deliver to their software relatively little computation per transistor.

[0028] Field Programmable Gate Arrays (FPGAs) are integrated circuits containing a large grid of general purpose digital elements with reconfigurable wiring between those elements. The elements originally were single digital gates, such as AND and OR gates, but evolved to larger elements that could, for instance, be programmed to map 6 inputs to 1 output according to any Boolean function. This architecture allows the FPGA to be configured from external sources to perform a wide variety of digital computations, which allows the device to be used as a co-processor to a CPU to accelerate computation. However, arithmetic operations such as multiplication and division on integers, and especially on floating point numbers, require many gates and can absorb a large fraction of an FPGA's general purpose resources. For this reason, modern FPGAs often devote a significant portion of their area to providing dozens or hundreds of multiplier blocks, which can be used instead of general purpose resources for computations requiring multiplication. These multiplier blocks typically perform 18 bit or wider integer multiplies, and use many transistors, as similar multiplier circuits do when they are part of a general purpose CPU.

[0029] Existing Field Programmable Analog Arrays (FPAAs) are analogous to FPGAs, but their configurable elements perform analog processing. These devices generally are intended to do signal processing, such as helping model neural circuitry. They are relatively low precision, have relatively low dynamic range, and introduce noise into computation. They have not been designed as, or intended for use as, general purpose computers. For instance, they are not seen by those having ordinary skill in the art as machines that can run the variety of complex algorithms with floating point arithmetic that typically run on high performance digital computers.

[0030] Finally, Graphics Processing Units (GPUs) are a variety of parallel processor that evolved to provide high speed graphics capabilities to personal computers. They offer standard floating point computing abilities with very high performance for certain tasks. Their computing model is sometimes based on having thousands of nearly identical threads of computing (SIMT), which are executed by a collection of SIMD-like internal computing engines, each of which is directed and redirected to perform work for which a slow external DRAM memory has provided data. Like other machines that implement standard floating point arithmetic, they use many transistors for that arithmetic. They are as wasteful of those transistors, in the sense discussed above, as are general purpose CPUs.

[0031] Some GPUs include support for 16 bit floating point values (sometimes called the "Half" format). The GPU manufacturers, currently such as NVIDIA or AMD/ATI, describe this capability as being useful for rendering images with higher dynamic range

than the usual 32 bit RGBA format, which uses 8 bits of fixed point data per color, while also saving space over using 32 bit floating point for color components. The special effects movie firm Industrial Light and Magic (ILM) independently defined an identical representation in their OpenEXR standard, which they describe as "a high dynamic-range (HDR) image file format developed by Industrial Light & Magic for use in computer imaging applications." Wikipedia (late 2008) describes the 16 bit floating point representation thusly: "This format is used in several computer graphics environments including OpenEXR, OpenGL, and D3DX. The advantage over 8-bit or 16-bit binary integers is that the increased dynamic range allows for more detail to be preserved in highlights and shadows. The advantage over 32-bit single precision binary formats is that it requires half the storage and bandwidth."

[0032] When a graphics processor includes support for 16 bit floating point, that support is alongside support for 32 bit floating point, and increasingly, 64 bit floating point. That is, the 16 bit floating point format is supported for those applications that want it, but the higher precision formats also are supported because they are believed to be needed for traditional graphics applications and also for so called "general purpose" GPU applications. Thus, existing GPUs devote substantial resources to 32 (and increasingly 64) bit arithmetic and are wasteful of transistors in the sense discussed above.

[0033] The variety of architectures mentioned above are all attempts to get more performance from silicon than is available in a traditional processor design. But designers of traditional processors also have been struggling to use the enormous increase in available transistors to improve performance of their machines. These machines often are required, because of history and economics, to support large existing instruction sets, such as the Intel x86 instruction set. This is difficult, because of the law of diminishing returns, which does not enable twice the performance to be delivered by twice the transistor count. One facet of these designers' struggle has been to increase the precision of arithmetic operations, since transistors are abundant and some applications could be sped up significantly if the processor natively supported long (e.g., 64 bit) numbers. With the increase of native fixed point precision from 8 to 16 to 32 to 64 bits, and of floating point from 32 to 64 and sometimes 128 bits, programmers have come to think in terms of high precision and to develop algorithms based on the assumption that computer processors provide such precision, since it comes as an integral part of each new generation of silicon chips and thus is "free."

[0034] Embodiments of the present invention efficiently provide computing power using a fundamentally different approach than those described above. In particular, embodiments of the present invention are directed to computer processors or other devices which use low precision high dynamic range (LPHDR) processing elements to perform computations (such as arithmetic operations).

[0035] One variety of LPHDR arithmetic represents values from one millionth up to one million with a precision of about 0.1%. If these values were represented and manipulated using the methods of floating point arithmetic, they would have binary mantissas of no more than 10 bits plus a sign bit and binary exponents of at least 5 bits plus a sign bit. However, the circuits to multiply and divide these floating point values would be relatively large. One example of an alternative embodiment is to use a logarithmic representation of the values. In such an approach, the values require the same number of bits to represent, but multiplication and division are implemented as addition and subtraction, respectively, of the logarithmic representations. Addition and



subtraction may be implemented efficiently as described below. As a result, the area of the arithmetic circuits remains relatively small and a greater number of computing elements can be fit into a given area of silicon. This means the machine can perform a greater number of operations per unit of time or per unit power, which gives it an advantage for those computations able to be expressed in the LPHDR framework.

[0036] Another embodiment is to use analog representations and processing mechanisms. Analog implementation of LPHDR arithmetic has the potential to be superior to digital implementation, because it tends to use the natural analog physics of transistors or other physical devices instead of using only the digital subset of the device's behavior. This fuller use of the devices' natural abilities may permit smaller mechanisms for doing LPHDR arithmetic. In recent years, in the field of silicon circuitry, analog methods have been supplanted by digital methods. In part, this is because of the ease of doing digital design compared to analog design. Also in part, it is because of the continued rapid scaling of digital technology ("Moore's Law") compared to analog technology. In particular, at deep submicron dimensions, analog transistors no longer work as they had in prior generations of larger-scale technology. This change of familiar behavior has made analog design still harder in recent years. However, digital transistors are in fact analog transistors used in a digital way, meaning digital circuits are really analog circuits designed to attempt to switch the transistors between completely on and completely off states. As scaling continues, even this use of transistors is starting to come face to face with the realities of analog behavior. Scaling of transistors for digital use is expected either to stall or to require digital designers increasingly to acknowledge and work with analog issues. For these reasons, digital embodiments may no longer be easy, reliable, and scalable, and analog embodiments of LPHDR arithmetic may come to dominate commercial architectures.

[0037] Because LPHDR processing elements are relatively small, a single processor or other device may include a very large number of LPHDR processing elements, adapted to operate in parallel with each other, and therefore may constitute a massively parallel LPHDR processor or other device. Such a processor or other device has not been described or practiced as a means of doing general purpose computing by those having ordinary skill in the art for at least two reasons. First, it is commonly believed by those having ordinary skill in the art, that LPHDR computation, and in particular massive amounts of LPHDR computation, whether performed in a massively parallel way or not, is not practical as a substrate for moderately general computing. Second, it is commonly believed by those having ordinary skill in the art that massive amounts of even high precision computation on a single chip or in a single machine, as is enabled by a compact arithmetic processing unit, is not useful without a corresponding increase in bandwidth between processing elements within the machine and into and out of the machine because computing is wire limited and arithmetic can be considered to be available at no cost.

[0038] Despite these views—that massive amounts of arithmetic on a chip or in a massively parallel machine are not useful, and that massive amounts of LPHDR arithmetic are even worse—embodiments of the present invention disclosed herein demonstrate that massively parallel LPHDR designs are in fact useful and provide significant practical benefits in at least several significant applications.

[0039] To conclude, modern digital computing systems provide high precision arithmetic, but that precision is costly. A modern double precision floating point multiplier may require on the order of a million transistors, even though only a handful

of transistors is required to perform a low precision multiplication. Despite the common belief among those having ordinary skill in the art that modern applications require high precision processing, in fact a variety of useful algorithms function adequately at much lower precision. As a result, such algorithms may be performed by processors or other devices implemented according to embodiments of the present invention, which come closer to achieving the goal of using a few transistors to multiply and a wire junction to add, thus enabling massively parallel arithmetic computation to be performed with relatively small amounts of physical resources (such as a single silicon chip). Although certain specialized tasks can function at low precision, it is not obvious, and in fact has been viewed as clearly false by those having ordinary skill in the art, that relatively general purpose computing such as is typically performed today on general purpose computers can be done at low precision. However, in fact a variety of useful and important algorithms can be made to function adequately at much lower than 32 bit precision in a massively parallel computing framework, and certain embodiments of the present invention support such algorithms, thereby offering much more efficient use of transistors, and thereby provide improved speed, power, and/or cost, compared to conventional computers.

[0040] Various computing devices implemented according to embodiments of the present invention will now be described. Some of these embodiments may be an instance of a SIMD computer architecture. Other architectures may be used, such as MIMD architectures, programmable array architectures (such as FPGAs and FPAAs), or GPU/SIMT architectures. The techniques disclosed herein may, for example, be implemented using any processor or other device having such an existing architecture, and replacing or augmenting some or all existing arithmetic units in the processor or other device, if any, with LPHDR arithmetic units in any of the ways disclosed herein. Devices implemented according to embodiments of the present invention, however, need not start with an existing processor design, but instead may be designed from scratch to include LPHDR arithmetic units within any of the architectures just described, or any other architecture.

[0041] Embodiments of the present invention may, for example, be implemented using the architecture of a particular kind of SIMD computer, the array processor. There are many variations and specific instances of array processors described in the scientific and commercial literature. Examples include the Illiac 4, the Connection Machine 1 and 2, the Goodyear MPP, and the MasPar line of computers.

[0042] Embodiments of the present invention need not, however, be implemented as SIMD computers. For example, embodiments of the present invention may be implemented as FPGAs, FPAAs, or related architectures that provide for flexible connectivity of a set of processing elements. For example, embodiments of the present invention may be implemented as GPU/SIMTs and as MIMDs, among others. For example, embodiments of the present invention may be implemented as any kind of machine which uses LPHDR arithmetic processing elements to provide computing using a small amount of resources (e.g., transistors or volume) compared with traditional architectures. Furthermore, references herein to "processing elements" within embodiments of the present invention should be understood more generally as any kind of execution unit, whether for performing LPHDR operations or otherwise.

[0043] An example SIMD computing system 100 is illustrated in FIG. 1. The computing system 100 includes a collection of many processing elements (PEs) 104. Sometimes present are a control unit (CU) 106, an I/O unit (IOU) 108, various Peripheral devices

110, and a Host computer 102. The collection of PEs is referred to herein as "the Processing Element Array" (PEA), even though it need not be two-dimensional or an array or grid or other particular layout. Some machines include additional components, such as an additional memory system called the "Staging Memory" in the Goodyear MPP, but these additional elements are neither essential in the computer nor needed to understand embodiments of the present invention and therefore are omitted here for clarity of explanation. One embodiment of the present invention is a SIMD computing system of the kind shown in FIG. 1, in which one or more (e.g., all) of the PEs in the PEA 104 are LPHDR elements, as that term is used herein.

[0044] The Host 102 is responsible for overall control of the computing system 100. It performs the serial, or mostly serial, computation typical of a traditional uni-processor. The Host 102 could have more complicated structure, of course, including parallelism of various sorts. Indeed a heterogeneous computing system combining multiple computing architectures in a single machine is a good use for embodiments of the present invention.

[0045] A goal of the Host 102 is to have the PEA 104 perform massive amounts of computation in a useful way. It does this by causing the PEs to perform computations, typically on data stored locally in each PE, in parallel with one another. If there are many PEs, much work gets done during each unit of time.

[0046] The PEs in the PEA 104 may be able to perform their individual computations roughly as fast as the Host 102 performs its computations. This means it may be inefficient to have the Host 102 attempt to control the PEA 104 on a time scale as fine as the Host's or PEA's minimal time step. (This minimal time, in a traditional digital design, would be the clock period.) For this reason, the specialized control unit (CU) 106 may be included in the architecture. The CU 106 has the primary task of retrieving and decoding instructions from an instruction memory, which conceptually is part of the CU 106, and issuing the partially decoded instructions to all the PEs in the PEA 104. (This may be viewed by the CU software as happening roughly simultaneously for all the PEs, though it need not literally be synchronous, and in fact it may be effective to use an asynchronous design in which multiple instructions at different stages of completion simultaneously propagate gradually across the PEA, for instance as a series of wave fronts.)

[0047] In a design which includes the CU 106, the Host 102 typically will load the instructions (the program) for the PEA 104 into the CU instruction memory (not shown in FIG. 1), then instruct the CU 106 to interpret the program and cause the PEA 104 to compute according to the instructions. The program may, for example, look generally similar to a typical machine language program, with instructions to cause data movement, logical operations, arithmetic operations, etc., in and between the PEs and other instructions to do similar operations together with control flow operations within the CU 106. Thus, the CU 106 may run a typical sort of program, but with the ability to issue massively parallel instructions to the PEA 104.

[0048] In order to get data into and out of the CU 106 and PEA 104, the I/O Unit 108 may interface the CU 106 and PEA 104 with the Host 102, the Host's memory (not shown in FIG. 1), and the system's Peripherals 110, such as external storage (e.g., disk drives), display devices for visualization of the computational results, and sometimes special high bandwidth input devices (e.g., vision sensors). The PEA's ability to process data far faster than the Host 102 makes it useful for the IOU 108 to be able to



completely bypass the Host 102 for some of its data transfers. Also, the Host 102 may have its own ways of communicating with the Peripherals 110.

[0049] The particular embodiment illustrated in FIG. 1 is shown merely for purposes of example and does not constitute a limitation of the present invention. For example, alternatively the functions performed by the CU 106 could instead be performed by the Host 102 with the CU 106 omitted. The CU 106 could be implemented as hardware distant from the PEA 104 (e.g., off-chip), or the CU 106 could be near to the PEA 104 (e.g., on-chip). I/O could be routed through the CU 106 with the IOU 108 omitted or through the separate I/O controller 108, as shown. Furthermore, the Host 102 is optional; the CU 106 may include, for example, a CPU, or otherwise include components sufficient to replace the functions performed by the Host 102. The Peripherals 110 shown in FIG. 1 are optional. The design shown in FIG. 1 could have a special memory, such as the Goodyear MPP's "staging memory," which provides an intermediate level of local storage. Such memory could, for example, be bonded to the LPHDR chip using 3D fabrication technology to provide relatively fast parallel access to the memory from the PEs in the PEA 104.

[0050] The PEA 104 itself, besides communicating with the CU 106 and IOU 108 and possibly other mechanisms, has ways for data to move within the array. For example, the PEA 104 may be implemented such that data may move from PEs only to their nearest neighbors, that is, there are no long distance transfers. FIGS. 2 and 3 show embodiments of the present invention which use this approach, where the nearest neighbors are the four adjacent PEs toward the North, East, West, and South, called a NEWS design. For example, FIG. 2 shows a subset of the PEs in PEA 104, namely PE 202, PE 204, PE 206, PE 208, and PE 210. When the CU 106 issues data movement instructions, all the PEs access data from or send data to their respective specified nearest neighbor. For instance, every PE might access a specified data value in its neighbor to the West and copy it into its own local storage. In some embodiments, such as some analog embodiments, these kinds of transfers may result in some degradation of the value copied.

[0051] FIG. 3 shows a PE 302 that includes data connections to the IOU 108. PE 302 is connected at the North to PE 304, at the East to PE 306, at the South to PE 308, and at the West to PE 310. However, driving signals from inside the PEA 104 out to the IOU 108 usually requires a physically relatively large driving circuit or analogous mechanism. Having those at every PE may absorb much of the available resources of the hardware implementation technology (such as VLSI area). In addition, having independent connections from every PE to the IOU 108 means many such connections, and long connections, which also may absorb much of the available hardware resources. For these reasons, the connections between the PEs and the IOU 108 may be limited to those PEs at the edges of the PE array 104. In this case, to get data out of, and perhaps into, the PEA 104, the data is read and written at the edges of the array and CU instructions are performed to shift data between the edges and interior of the PEA 104. The design may permit data to be pushed from the IOU 108 inward to any PE in the array using direct connections, but may require readout to occur by using the CU 106 to shift data to the edges where it can be read by the IOU 108.

[0052] Connections between the CU 106 and PEA 104 have analogous variations. One design may include the ability to drive instructions into all the PEs roughly simultaneously, but another approach is to have the instructions flow gradually (for

instance, shift in discrete time steps) across the PEA 104 to reach the PEs. Some SIMD designs, which may be implemented in embodiments of the present invention, have a facility by which a "wired-or" or "wired-and" of the state of every PE in the PEA 104 can be read by the CU 106 in approximately one instruction delay time.

[0053] There are many well studied variations on these matters in the literature, any of which may be incorporated into embodiments of the present invention. For example, an interconnect, such as an 8-way local interconnect, may be used. The local connections may include a mixture of various distance hops, such as distance 4 or 16 as well as distance 1. The outside edges may be connected using any topology, such as a torus or twisted torus. Instead of or in addition to a local interconnect, a more complex global interconnect, such as the hypercube design, may be used. Furthermore, the physical implementation of the PEA 104 (e.g., a chip) could be replicated (e.g., tiled on a circuit board) to produce a larger PEA. The replication may form a simple grid or other arrangement, just as the component PEAs may but need not be grids.

[0054] FIG. 4 shows an example design for a PE 400 (which may be used to implement any one or more of the PEs in the PEA 104). The PE 400 stores local data. The amount of memory for the local data varies significantly from design to design. It may depend on the implementation technologies available for fabricating the PE 400. Sometimes rarely changing values (Constants) take less room than frequently changing values (Registers), and a design may provide more Constants than Registers. For instance, this may be the case with digital embodiments that use single transistor cells for the Constants (e.g., floating gate Flash memory cells) and multiple transistor cells for the Registers (e.g., 6-transistor SRAM cells). Sometimes the situation is reversed, as may be the case in analog embodiments, where substantial area for capacitance may be needed to ensure stable long term storage of Constants, and such embodiments may have more Registers than Constants. Typical storage capacities might be tens or hundreds of arithmetic values stored in the Registers and Constants in each PE, but these capacities are adjustable by the designer. Some designs, for instance, may have Register storage but no Constant storage. Some designs may have thousands or even many more values stored in each PE. All of these variations may be reflected in embodiments of the present invention.

[0055] Each PE needs to operate on its local data. For this reason within the PE 400 there are data paths 402a-i, routing mechanisms (such as the multiplexor MUX 404), and components to perform some collection of logical and arithmetic operations (such as the logic unit 406 and the LPHDR arithmetic unit 408). The LPHDR arithmetic unit 408 performs LPHDR arithmetic operations, as that term is used herein. The input, output, and intermediate "values" received by, output by, and operated on by the PE 400 may, for example, take the form of electrical signals representing numerical values.

[0056] The PE 400 also may have one or more flag bits, shown as Mask 410 in FIG. 4. The purpose of the Mask 410 is to enable some PEs, the ones in which a specified Mask bit is set, to ignore some instructions issued by the CU 106. This allows some variation in the usual lock-step behaviors of all PEs in the PEA 104. For instance, the CU 106 may issue an instruction that causes each PE to reset or set its Mask 410 depending on whether a specified Register in the PE is positive or negative. A subsequent instruction, for instance an arithmetic instruction, may include a bit meaning that the instruction should be performed only by those PEs whose Mask 410 is reset. This combination has the effect of conditionally performing the arithmetic instruction in each PE depending on whether the specified Register in that PE was

positive. As with the Compare instructions of traditional computers, there are many possible design choices for mechanisms to set and clear Masks.

[0057] The operation of the PEs is controlled by control signals 412a-d received from the CU 106, four of which are shown in FIG. 4 merely for purposes of example and not limitation. We have not shown details of this mechanism, but the control signals 412a-d specify which Register or Constant memory values in the PE 400 or one of its neighbors to send to the data paths, which operations should be performed by the Logic 406 or Arithmetic 408 or other processing mechanisms, where the results should be stored in the Registers, how to set, reset, and use the Mask 410, and so on. These matters are well described in the literature on SIMD processors.

[0058] Many variations of this PE 400 and PEA design are possible and fall within the scope of the present invention. Digital PEs can have shifters, lookup tables, and many other mechanisms such as described in the literature. Analog PEs can have time-based operators, filters, comparators with global broadcast signals and many other mechanisms such as described in the literature. The PEA 104 can include global mechanisms such as wired-OR or wired-AND for digital PEAs or wired-SUM for analog PEAs. Again, there are many variations well described in the literature on digital and analog computing architectures.

[0059] For example, LPHDR operations other than and/or in addition to addition and multiplication may be supported. For example, a machine which can only perform multiplication and the function  $(1-X)$  may be used to approximate addition and other arithmetic operations. Other collections of LPHDR operations may be used to approximate LPHDR arithmetic operations, such as addition, multiplication, subtraction, and division, using techniques that are well-known to those having ordinary skill in the art.

[0060] One aspect of embodiments of the present invention that is unique is the inclusion of LPHDR arithmetic mechanisms in the PEs. Embodiments of such mechanisms will now be described.

[0061] One digital embodiment of the LPHDR arithmetic unit 408 operates on digital (binary) representations of numbers. In one digital embodiment these numbers are represented by their logarithms. Such a representation is called a Logarithmic Number System (LNS), which is well-understood by those having ordinary skill in the art.

[0062] In an LNS, numbers are represented as a sign and an exponent. There is an implicit base for the logarithms, typically 2 when working with digital hardware. In the present embodiment, a base of 2 is used for purposes of example. As a result, a value, say B, is represented by its sign and a base 2 logarithm, say b, of its absolute value. For numbers to have representation errors of at most, say, 1% (one percent), the fractional part of this logarithm should be represented with enough precision that the least possible change in the fraction corresponds to about a 1% change in the value B. If fractions are represented using 6 bits, increasing or decreasing the fraction by 1 corresponds to multiplying or dividing B by the 64th root of 2, which is approximately 1.011. This means that numbers may be represented in the present embodiment with a multiplicative error of approximately 1%. So, in this example embodiment the fraction part of the representation has 6 bits.

[0063] Furthermore, the space of values processed in the present embodiment have



high dynamic range. To represent numbers whose absolute value is from, say, one billionth to one billion, the integer part of the logarithm must be long enough to represent plus or minus the base 2 logarithm of one billion. That logarithm is about 29.9. In the present embodiment the integer part of the logarithm representation is 5 bits long to represent values from 0 to 31, which is sufficient. There also is a sign bit in the exponent. Negative logarithms are represented using two's complement representation.

[0064] In an LNS, the value zero corresponds to the logarithm negative infinity. One can choose a representation to explicitly represent this special value. However, to minimize resources (for instance, area) used by arithmetic circuits, the present embodiment represents zero by the most negative possible logarithm, which is  $-32$ , corresponding to the two's complement bit representation '100000 000000', and denoting a value of approximately  $2.33\text{E}-10$ .

[0065] When computing, situations can arise in which operations cannot produce reasonable values. An example is when a number is too large to be represented in the chosen word format, such as when multiplying or adding two large numbers or upon divide by zero (or nearly zero). One common approach to this problem is to allow a value to be marked as Not A Number (NaN) and to make sure that each operation produces NaN if a problem arises or if either of its inputs is NaN. The present embodiment uses this approach, as will be described in the following.

[0066] FIG. 5 shows the word format 500 for these numbers, in the present embodiment. It has one NaN bit 502a, one bit 502b for the sign of the value, and 12 bits 502c-e representing the logarithm. The logarithm bits include a 5 bit integer part 502d and a 6 bit fraction part 502e. To permit the logarithms to be negative, there is a sign bit 502c for the logarithm which is represented in two's complement form. The NaN bit is set if some problem has arisen in computing the value. The word format 500 shown in FIG. 5 is merely an example and does not constitute a limitation of the present invention. Other variations may be used, so long as they have low precision and high dynamic range.

[0067] FIG. 6 shows an example digital implementation of the LPHDR arithmetic unit 408 for the representation illustrated in FIG. 5. The unit 408 receives two inputs, A 602a and B 602b, and produces an output 602c. The inputs 602a-b and output 602c may, for example, take the form of electrical signals representing numerical values according to the representation illustrated in FIG. 5, as is also true of signals transmitted within the unit 408 by components of the unit 408. The inputs 602a-b and output 602c each are composed of a Value and a NaN (Not A Number) bit. The unit 408 is controlled by control signals 412a-d, coming from the CU 106, that determine which available arithmetic operation will be performed on the inputs 602a-b. In this embodiment, all the available arithmetic operations are performed in parallel on the inputs 602a-b by adder/subtractor 604, multiplier 606, and divider 608. Adder/subtractor 604 performs LPHDR addition and subtraction, multiplier 606 performs LPHDR multiplication, and divider 608 performs LPHDR division.

[0068] The desired result (from among the outputs of adder/subtractor 604, multiplier 606, and divider 608) is chosen by the multiplexers (MUXes) 610a and 610b. The right hand MUX 610b sends the desired value to the output 602c. The left hand MUX 610a sends the corresponding NaN bit from the desired operation to the OR gate 612, which outputs a set NaN bit if either input is NaN or if the specified arithmetic operation yields

NAN. The computing architecture literature discusses many variations which may be incorporated into the embodiment illustrated in FIG. 6.

[0069] LNS arithmetic has the great advantage that multiplication (MUL) and division (DIV) are very easy to compute and take few physical resources (e.g., little area in a silicon implementation). The sign of the result is the exclusive-or of the signs of the operands. The logarithm part of the output is the sum, in the case of MUL, or the difference, in the case of DIV, of the logarithm parts of the operands. The sum or difference of the logarithms can overflow, producing a NAN result. Certain other operations also are easy in LNS arithmetic. For instance, square root corresponds to dividing the logarithm in half, which in our representation means simply shifting it one bit position to the right.

[0070] Thus, the multiplier 606 and divider 608 in FIG. 6 are implemented as circuits that simply add or subtract their inputs, which are two's complement binary numbers (which in turn happen to be logarithms). If there is overflow, they output a 1 for NAN.

[0071] Implementing addition and subtraction in LNS, that is, the adder/subtractor 604 in FIG. 6, follows a common approach used in the literature on LNS. Consider addition. If we have two positive numbers B and C represented by their logarithms b and c, the representation of the sum of B and C is  $\log(B+C)$ . An approach to computing this result that is well known to those skilled in the art is based on noticing that  $\log(B+C) = \log(B * (1+C/B)) = \log(B) + \log(1+C/B) = b + F(c-b)$  where  $F(x) = \log(1+2^x)$ . Thus, the present embodiment computes c-b, feeds that through F, and adds the result to b, using standard digital techniques known to those skilled in the art.

[0072] Much of the published literature about LNS is concerned with how to compute F(x), the special function for ADD, along with a similar function for SUB. Often these two functions share circuitry, and this is why a single combined adder/subtractor 604 is used in the embodiment of FIG. 6. There are many published ways to compute these functions or approximations to them, including discussions of how to do this when the values are of low precision. Any such method, or other method, may be used. Generally speaking, the more appropriate variations for massively parallel LPHDR arithmetic are those that require the minimal use of resources, such as circuit area, taking advantage of the fact that the representation used in the embodiment of FIG. 6 is low precision and that the arithmetic operations need not be deterministic nor return the most accurate possible answer within the low precision representation. Thus, embodiments of the present invention may use circuitry that does not compute the best possible answer, even among the limited choices available in a low precision representation.

[0073] In order to enable conditional operation of selected PEs, the present embodiment is able to reset and set the MASK flag 410 based on results of computations. The mechanism for doing this is that the CU 106 includes instructions that cause the MASK 410 in each PE to unconditionally reset or set its flag along with other instructions to perform basic tests on values entering the MASK 410 on data path 402f and to set the flag accordingly. Examples of these latter instructions include copying the sign bit or NAN bit of the word on data path 402f into the MASK bit 410. Another example is to set the MASK bit 410 if the 12 bit value part of the word on data path 402f is equal to binary zero. There are many additional and alternative ways for doing this that are directly analogous to comparison instructions in traditional processors and which are well understood by those skilled in the art.



[0074] It is worth noting that while the obvious method of using the above LNS operations is to do LPHDR arithmetic, the programmer also may consider selected values to be 12 bit two's complement binary numbers. MUL and DIV may be used to add and subtract such values, since that is precisely their behavior in LNS implementations. The Mask setting instructions can compare these simple binary values. So besides doing LPHDR computations, this digital embodiment using LNS can perform simple binary arithmetic on short signed integers.

[0075] Some embodiments of the present invention may include analog representations and processing methods. Such embodiments may, for example, represent LPHDR values as charges, currents, voltages, frequencies, pulse widths, pulse densities, various forms of spikes, or in other forms not characteristic of traditional digital implementations. There are many such representations discussed in the literature, along with mechanisms for processing values so represented. Such methods, often called Analog methods, can be used to perform LPHDR arithmetic in the broad range of architectures we have discussed, of which SIMD is one example.

[0076] An example of an analog SIMD architecture is the SCAMP design (and related designs) of Dudek. In that design values have low dynamic range, being accurate roughly to within 1%. Values are represented by charges on capacitors. Those capacitors typically are the gates of transistors. Each PE has several memory cells, analogous to the Registers shown in FIG. 4. Addition is performed by turning on pass transistors from the two operands, which transfer their charge onto an analog bus, where it is summed by the natural physics of charge and wires, upon which it is gated onto another Register to charge up its capacitor, which then represents the sum of the operands. The detailed mechanism disclosed by Dudek actually produces the negative of the sum, but the basic concept is as described and is a simple way to perform addition and subtraction using analog representations and simple processing mechanisms.

[0077] Variations of the SCAMP design have been fabricated and used to perform a range of low precision, low dynamic range computations related to image processing. These designs do not perform high dynamic range arithmetic, nor do they include mechanisms for performing multiplication or division of values stored in Registers. However, the Dudek designs suggest the general feasibility of constructing analog SIMD machines. The following describes how to build an analog SIMD machine that performs LPHDR arithmetic, and is thus an embodiment of the present invention.

[0078] One embodiment of the present invention represents values as a mixture of analog and digital forms. This embodiment represents values as low precision, normalized, base 2 floating point numbers, where the mantissa is an analog value and the exponent is a binary digital value. The analog value may be accurate to about 1%, following the approach of Dudek, which is well within the range of reasonable analog processing techniques. The exponent may be 6 bits long, or whatever is needed to provide the desired high dynamic range.

[0079] To multiply values, the embodiment proceeds by analogy to traditional floating point methods. The digital exponents are summed using a binary arithmetic adder, a standard digital technique. The analog mantissas are multiplied. Since they represent normalized values between approximately  $1/2$  and 1, their product may be as small as approximately  $1/4$ . Such a product value needs to be normalized back to the range  $1/2$

to 1. This is done, in the present embodiment, by comparing the analog mantissa to an analog representation of  $1/2$ , using a threshold circuit. If the mantissa is below  $1/2$ , then it is doubled and one is subtracted from the exponent, where such subtraction is simple digital subtraction. Doubling the mantissa is implemented in a way that corresponds to the chosen analog representation. For example, whatever means are being used to add two analog values can be used to double the mantissa, by adding it to a copy of itself. For example, if the mantissa is represented as a current, such as copy may be produced by a current mirror, or other suitable mechanism, and addition may be performed by a current summing junction.

[0080] The means of multiplying the original analog mantissas depends on the representation chosen. For example, if mantissas are represented using charge, following SCAMP, then any known method from the literature may be used to convert charge to current. For instance, since the charge on a capacitor determines the voltage on the capacitor, this may be implemented as a conversion from voltage to current, which is a basic technique in analog electronics known to those skilled in the art. In any case, if the mantissas are represented as currents, or once the mantissas are converted to currents, they may be multiplied using, for instance, the techniques of Gilbert. The Gilbert multiplier produces a current, representing the product, which may, if necessary, then be converted back to charge (or whatever representation is used). These are merely examples of how the needed operations might be performed. The literature discusses these matters extensively and these kinds of analog circuits are known to those skilled in the art.

[0081] Adding and subtracting values requires pre-normalization of the values to the same exponent, as is done in traditional digital floating point arithmetic. The present embodiment does this by comparing the exponents and choosing the smaller one. Then the smaller one is subtracted from the larger, using digital means. The difference specifies how many times the mantissa which corresponds to the smaller exponent needs to be divided in half. If that mantissa is represented by (or converted to) a current, then an analog R-2R style ladder may be used to divide the current in half the required number of times, with the stage of the ladder specified by the difference of exponents calculated as above. The resulting scaled down current is added to (or subtracted from, if this is an LPHDR subtraction operation) the current corresponding to the mantissa associated with the larger exponent to yield the output mantissa. The output exponent associated with the output mantissa is the larger exponent. Post-normalization may be needed at this point. If the output mantissa is greater than 1, then it needs to be divided in half and the output exponent needs to be incremented. If it is less than  $1/2$ , then it needs to be doubled enough times to exceed  $1/2$  and the output exponent must be decremented correspondingly, which may be performed by a series of threshold circuits, doubler circuits, and associated decrementer circuits. These increments and decrements of the binary digital exponent, and corresponding doublings and halvings of the analog mantissa current, are straightforward operations well known to those skilled in the art.

[0082] The present embodiment represents the exponent as a digital binary number. Alternate embodiments may represent the exponent as an analog value. However, it is important that the exponent be represented in storage and computation in such a manner that neither noise nor other errors cause a change in the value it represents. Such changes in the exponent could introduce factors of two (or in some embodiments larger) changes in the values of the stored numbers. To maintain accuracy of the exponents, an embodiment may quantize the exponent to relatively few levels, for

instance 16 values plus a sign bit. During processing, slight variations in the analog representation of the exponent may then be removed by circuitry that restores values to the 16 standard quantization levels. To get sufficient dynamic range in such an embodiment, the floating point numbers may be processed as base 4 numbers, rather than the usual base 2 numbers. This means, for instance, that normalized mantissas are in the range  $1/4$  to 1. The methods discussed above for addition, subtraction, and multiplication apply as described, with slight and straightforward variations.

[0083] The analog and mixed signal embodiments discussed above are merely examples and do not constitute a limitation of the present invention. The published literature on neuromorphic, analog, and mixed signal techniques provides a wealth of methods that enable LPHDR storage and processing to be implemented. Such storage and processing may introduce noise as well as fabrication errors into the behavior of machines performing LPHDR arithmetic. The results we present below, on software applications running using "fp+noise" arithmetic, show that despite these very "un-digital" qualities a machine built in this way is surprisingly useful.

[0084] Evidence that LPHDR arithmetic is useful in several important practical computing applications will now be provided. The evidence is presented for a broad variety of embodiments of the present invention, thereby showing that the usefulness does not depend much on the detailed implementation.

[0085] For the goal of showing usefulness, we choose a very general embodiment of an LPHDR machine. Our model of the machine is that it provides at least the following capabilities: (1) is massively parallel, (2) provides LPHDR arithmetic possibly with noise, (3) provides a small amount of memory local to each arithmetic unit, (4) provides the arithmetic/memory units in a two-dimensional physical layout with only local connections between units (rather than some more powerful, flexible, or sophisticated connection mechanism), and (5) provides only limited bandwidth between the machine and the host machine. Note that this model is merely an example which is used for the purpose of demonstrating the utility of various embodiments of the present invention, and does not constitute a limitation of the present invention. This model includes, among others, implementations that are digital or analog or mixed, have zero or more noise, have architectures which are FPGA-like, or SIMD-like, or MIMD-like, or otherwise meet the assumptions of the model. More general architectures, such as shared memory designs, GPU-like designs, or other sophisticated designs subsume this model's capabilities, and so LPHDR arithmetic in those architectures also is useful. While we are thus showing that LPHDR arithmetic is useful for a broad range of designs, of which SIMD is only an instance, for purpose of discussion below, we call each unit, which pairs memory with arithmetic, a Processing Element or "PE".

[0086] Several applications are discussed below. For each, the discussion shows (1) that the results are useful when computation is performed in possibly noisy LPHDR arithmetic, and (2) that the computation can be physically laid out in two dimensions with only local flow of data between units, only limited memory within each unit, and only limited data flow to/from the host machine, in such a way that the computation makes efficient use of the machine's resources (area, time, power). The first requirement is referred to as "Accuracy" and the second requirement "Efficiency." Applications that meet both requirements running in this model will function well on many kinds of LPHDR machines, and thus those machines are a broadly useful invention.



[0087] Applications are tested using two embodiments for the machine's arithmetic. One uses accurate floating point arithmetic but multiplies the result of each arithmetic operation by a uniformly chosen random number between 0.99 and 1.01. In the following discussion, this embodiment is denoted "fp+noise". It may represent the results produced by an analog embodiment of the machine.

[0088] A second embodiment uses logarithmic arithmetic with a value representation as shown in FIG. 5. The arithmetic is repeatable, that is, not noisy, but because of the short fraction size it produces errors of up to approximately 1-2% in each operation. In the following discussion, this embodiment is denoted "lms". It may represent the results produced by a particular digital embodiment of the machine.

[0089] To demonstrate usefulness of embodiments of the invention, we shall discuss three computational tasks that are enabled by embodiments of the invention and which in turn enable a variety of practical applications. Two of the tasks are related to finding nearest neighbors and the other is related to processing visual information. We shall describe the tasks, note their practical application, and then demonstrate that each task is solvable using the general model described above and thus solvable using embodiments of the present invention.

#### Application 1

##### Finding Nearest Neighbors

[0090] Given a large set of vectors, called Examples, and a given vector, called Test, the nearest neighbor problem ("NN") is to find the Example which is closest to Test where the distance metric is the square of the Euclidean distance (sum of squares of distances between respective components).

[0091] NN is a widely useful computation. One use is for data compression, where it is called "vector quantization". In this application we have a set of relatively long vectors in a "code book" (these are the Examples) and associated short code words (for instance the index of the vector in the code book). We move through a sequence of vectors to be compressed, and for each such vector (Test), find the nearest vector in the code book and output the corresponding code word. This reduces the sequence of vectors to the shorter sequence of code words. Because the code words do not completely specify the original sequence of vectors, this is a lossy form of data compression. Among other applications, it may be used in speech compression and in the MPEG standards.

[0092] Another application of NN would be in determining whether snippets of video occur in a large video database. Here we might abstract frames of video from the snippet into feature vectors, using known methods, such as color histograms, scale invariant feature extraction, etc. The Examples would be analogous feature vectors extracted from the video database. We would like to know whether any vector from the snippet was close to any vector from the database, which NN can help us decide.

[0093] In many applications of nearest neighbor, we would prefer to find the true nearest neighbor but it is acceptable if we sometimes find another neighbor that is only slightly farther away or if we almost always find the true nearest neighbor. Thus, an approximate solution to the nearest neighbor problem is useful, especially if it can be computed especially quickly, or at low power, or with some other advantage compared

to an exact solution.

[0094] We shall now show that approximate nearest neighbor is computable using embodiments of the present invention in a way that meets the criteria of Accuracy and Efficiency.

[0095] Algorithm. The following describes an algorithm which may be performed by machines implemented according to embodiments of the present invention, such as by executing software including instructions for performing the algorithm. The inputs to the algorithm are a set of Examples and a Test vector. The algorithm seeks to find the nearest (or almost nearest) Example to the Test.

[0096] In the simplest version of the algorithm, the number of Examples may be no larger than the number of PEs and each vector must be short enough to fit within a single PE's memory. The Examples are placed into the memories associated with the PEs, so that one Example is placed in each PE. Given a Test, the Test is passed through all the PEs, in turn. Accompanying the Test as it passes through the PEs is the distance from the Test to the nearest Example found so far, along with information that indicates what PE (and thus what Example) yielded that nearest Example found so far. Each PE computes the distance between the Test and the Example stored in that PE's memory, and then passes along the Test together with either the distance and indicator that was passed into this PE (if the distance computed by this PE exceeded the distance passed into the PE) or the distance this PE computed along with information indicating this PE's Example is the nearest so far (if the distance computed by this PE is less than the distance passed into the PE). Thus, the algorithm is doing a simple minimization operation as the Test is passed through the set of PEs. When the Test and associated information leave the last PE, the output is a representation of which PE (and Example) was closest to the Test, along with the distance between that Example and the Test.

[0097] In a more efficient variant of this algorithm, the Test is first passed along, for example, the top row, then every column passes the Test and associated information downward, effectively doing a search in parallel with other columns, and once the information reaches the bottom it passes across the bottom row computing a minimum distance Example of all the columns processed so far as it passes across the row. This means that the time required to process the Test is proportional to (the greater of) the number of PEs in a row or column.

[0098] An enhancement of this algorithm proceeds as above but computes and passes along information indicating both the nearest and the second nearest Example found so far. When this information exits the array of PEs, the digital processor that is hosting the PE array computes (in high precision) the distance between the Test and the two Examples indicated by the PE array, and the nearer of the two is output as the likely nearest neighbor to the Test.

[0099] Accuracy. We expressed the arithmetic performed by the enhanced algorithm described above as code in the C programming language. That code computes both nearest neighbors, which are discussed here, along with weighted scores, which are discussed below.

[0100] The C code performs the same set of arithmetic operations in the same order using the same methods of performing arithmetic as an actual implementation of the



present invention, such as one implemented in hardware. It thus yields the same results as the enhanced algorithm would yield when running on an implementation of the present invention. (How the algorithm is organized to run efficiently on such an implementation is discussed below in the section on Efficiency.)

[0101] In particular, when computing the distance between the Test and each Example, the code uses Kahan's method, discussed below, to perform the possibly long summation required to form the sum of the squares of the distances between vector components of the Test and Example.

[0102] The C code contains several implementations for arithmetic, as discussed above. When compiled with "#define fp" the arithmetic is done using IEEE standard floating point. If a command line argument is passed in to enable noisy arithmetic, then random noise is added to the result of every calculation. This is the "fp+noise" form of arithmetic. When compiled without "#define fp" the arithmetic is done using low precision logarithmic arithmetic with a 6 bit base-2 fraction. This is the "lns" form of arithmetic.

[0103] When the code was run it produced traces showing the results of the computations it performed. These traces, shown below, show that with certain command line arguments the enhanced algorithm yielded certain results for LPHDR nearest neighbor calculations. These results provide details showing the usefulness of this approach. We shall discuss the results briefly here.

[0104] The first results are for "fp+noise". Ten distinct runs were performed. Each run generated one million random Example vectors of length five, where each component of each vector was drawn from  $N(0,1)$ —the Gaussian (normal) distribution with mean zero and standard deviation 1. Each run then generated one hundred Test vectors of length five, where each component of each vector also was drawn from  $N(0,1)$ . For each Test, the nearest neighbor was computed both according to the enhanced algorithm above and according to the standard nearest neighbor method using high precision floating point arithmetic. A count was kept of the number of times the enhanced algorithm yielded the same result as the standard floating point method. The results were as follows:

```
%.a.out 5 10 1000000 100 1
```

Representation is Floating Point with noise.

Run 1. On 100 tests, 100(100.0%) matches and 0.81% mean score error.

Run 2. On 100 tests, 100(100.0%) matches and 0.84% mean score error.

Run 3. On 100 tests, 100(100.0%) matches and 0.98% mean score error.

Run 4. On 100 tests, 100(100.0%) matches and 0.81% mean score error.

Run 5. On 100 tests, 100(100.0%) matches and 0.94% mean score error.

Run 6. On 100 tests, 100(100.0%) matches and 0.82% mean score error.

Run 7. On 100 tests, 100(100.0%) matches and 0.78% mean score error.

Run 8. On 100 tests, 100(100.0%) matches and 0.86% mean score error.

Run 9. On 100 tests, 100(100.0%) matches and 0.85% mean score error.

Run 10. On 100 tests, 99(99.0%) matches and 0.86% mean score error.

Average percentage of time LPHDR (with final DP correction) finds nearest example=99.90%.

Average score error between LPHDR and DP=0.85%.

[0119] The "mean score error" values are considered below in the discussion of

weighted scores. The "matches" information is relevant here.

[0120] Of the ten runs, only one had any test, of the 100 tests performed, which yielded a nearest neighbor different from what the usual high precision method yielded. Thus, the average percentage of matches between the enhanced algorithm running with "fp+noise" arithmetic and the usual method was 99.9%.

[0121] A similar computation was then performed using "ins" arithmetic. In this case, the results were:

```
%.a.out 5 10 1000000 100 0
```

Representation is LNS without noise.

Run 1. On 100 tests, 100(100.0%) matches and 0.15% mean score error.

Run 2. On 100 tests, 100(100.0%) matches and 0.07% mean score error.

Run 3. On 100 tests, 100(100.0%) matches and 0.08% mean score error.

Run 4. On 100 tests, 100(100.0%) matches and 0.09% mean score error.

Run 5. On 100 tests, 100(100.0%) matches and 0.11% mean score error.

Run 6. On 100 tests, 100(100.0%) matches and 0.16% mean score error.

Run 7. On 100 tests, 100(100.0%) matches and 0.07% mean score error.

Run 8. On 100 tests, 100(100.0%) matches and 0.13% mean score error.

Run 9. On 100 tests, 99(99.0%) matches and 0.17% mean score error.

Run 10. On 100 tests, 98(98.0%) matches and 0.16% mean score error.

Average percentage of time LPHDR (with final DP correction) finds nearest example=99.70%.

Average score error between LPHDR and DP=0.12%.

[0136] The average percentage of matches was 99.7%, slightly worse than for "fp+noise".

[0137] The accuracy shown by the enhanced nearest neighbor algorithm using two forms of LPHDR arithmetic is surprising. To perform many calculations sequentially with 1% error and yet produce a final result with less than 1% error may seem counter-intuitive. Nonetheless, the LPHDR arithmetic proves effective, and the accuracy shown is high enough to be useful in applications for which approximate nearest neighbor calculations are useful.

[0138] As an extreme case, a variant of fp+noise was tested in which the noise varied uniformly from +10% to -5%. Thus, each arithmetic operation produced a result that was between 10% too large and 5% too small. The enhanced nearest neighbor algorithm, as described above, was performed where each run generated 100,000 Example vectors. The surprising results, below, show that even with this extreme level of imprecise, noisy, and non-zero mean LPHDR arithmetic, useful results can be achieved.

[0000] Run 1. On 100 tests, 97(97.0%) matches.

Run 2. On 100 tests, 100(100.0%) matches.

Run 3. On 100 tests, 100(100.0%) matches.

Run 4. On 100 tests, 98(98.0%) matches.

Run 5. On 100 tests, 98(98.0%) matches.

Run 6. On 100 tests, 99(99.0%) matches.

Run 7. On 100 tests, 99(99.0%) matches.

Run 8. On 100 tests, 99(99.0%) matches.

Run 9. On 100 tests, 99(99.0%) matches.

Run 10. On 100 tests, 99(99.0%) matches.

Average percentage of time LPHDR (with final DP correction) finds nearest example=98.80%.

[0139] Efficiency. In contrast to the surprising Accuracy results, it is clear to those having ordinary skill in the art that the calculations of the enhanced nearest neighbor algorithm can be performed efficiently in the computing model presented, where the arithmetic/memory units are connected in a two-dimensional physical layout, using only local communication between PEs. However, this does not address the matter of keeping the machine busy doing useful work using only low bandwidth to the host machine.

[0140] When computing the nearest neighbor to a single Test, the Test flows across all the PEs in the array. As discussed above, if the array is an  $M \times M$  grid, it takes at least  $O(M)$  steps for the Test to pass through the machine and return results to the host. During this time the machine performs  $O(M \times M)$  nearest neighbor distance computations, but since the machine is capable of performing  $O(M \times M)$  calculations at each step, a factor of  $O(M)$  is lost.

[0141] This speedup, compared to a serial machine, of a factor of  $O(M)$  is significant and useful. However, the efficiency can be even higher. If sufficiently many Test vectors, say  $O(M)$ , or more, are to be processed then they can be streamed into the machine and made to flow through in a pipelined fashion. The time to process  $O(M)$  Tests remains  $O(M)$ , the same as for a single Test, but now the machine performs  $O(M) \times O(M \times M)$  distance computations, and thus within a constant factor the full computing capacity of the machine is used.

[0142] Thus, the machine is especially efficient if it is processing at least as many Test vectors as the square root of the number of PEs. There are applications that fit well into this form, such as pattern recognition or compression of many independent Tests (e.g., blocks of an image, parts of a file, price histories of independent stocks) as well as the problem of finding the nearest neighbor to every Example in the set of Examples. This is in contrast to the general view among those having ordinary skill in the art, as discussed above, that machines with very many arithmetic processing elements on a single chip, or similar, are not very useful.

## Application 2

### Distance Weighted Scoring

[0143] A task related to Nearest Neighbor is Distance Weighted Scoring. In this task, each Example has an associated Score. This is a number that in some way characterizes the Example. For instance, if the Examples are abstractions of the history of prices of a given stock, the Scores might be historical probabilities of whether the price is about to increase or decrease. Given a Test vector, the task is to form a weighted sum of the Scores of all the Examples, where the weights are a diminishing function of the distance from the Test to the respective Examples. For example, this weighted score might be taken as a prediction of the future price of the stock whose history is represented by the Test. This use of embodiments of the invention might help support, for instance, high speed trading of stocks, as is performed by certain "quantitative" hedge funds, despite the general view by those having ordinary skill in



the art that low precision computation is not of use in financial applications.

[0144] The C code described above computes weighted scores along with nearest neighbors. The scores assigned to Examples in this computation are random numbers drawn uniformly from the range [0,1]. The weight for each Example in this computation is defined to be the un-normalized weight for the Example divided by the sum of the un-normalized weights for all Examples, where the un-normalized weight for each Example is defined to be the reciprocal of the sum of one plus the squared distance from the Example to the Test vector. As discussed above, the code performs a number of runs, each producing many Examples and Tests, and compares results of traditional floating point computations with results calculated using fp+noise and Ins arithmetic.

[0145] Looking again at the trace results of running the simulation, above, we see that for fp+noise the LPHDR weighted scores on average were within 0.85% of the correct value and never were as much as 1% different. For Ins arithmetic the errors were even smaller, averaging just 0.12% error.

[0146] These results are surprising given that computing an overall weighted score involves summing the individual weighted scores associated with each Example. Since each run was processing 1,000,000 Examples, this means that the sums were over one million small positive values. The naive method of summing one million small values with errors of about 1% in each addition should yield results that approximate noise. However, the code performs its sums using a long known method invented by Kahan (Kahan, William (January 1965), "Further remarks on reducing truncation errors", Communications of the ACM 8 (1): 40). The method makes it feasible to perform long sums, such as are done for Distance Weighted Scores, or as might be used in computational finance when computing prices of derivative securities using Monte Carlo methods, or for performing deconvolution in image processing algorithms, as will be discussed next.

[0147] The Efficiency of this algorithm is similar to that of NN, as discussed earlier. If many Test vectors are processed at once, the machine performs especially efficiently.

### Application 3

#### Removing Motion Blur in Images

[0148] In order to gather sufficient light to form an image, camera shutters are often left open for long enough that camera motion can cause blurring. This can happen as a result of camera shake in inexpensive consumer cameras as well as with very expensive but fast moving cameras mounted on satellites or aircraft. If the motion path of the camera is known (or can be computed) then the blur can be substantially removed using various deblurring algorithms. One such algorithm is the Richardson-Lucy method ("RL"), and we show here that embodiments of the present invention can run that algorithm and produce useful results. Following the discussion format above, we discuss criteria of Accuracy and Efficiency.

[0149] Algorithm. The Richardson-Lucy algorithm is well known and widely available. Assume that an image has been blurred using a known kernel. In particular, assume that the kernel is a straight line and that the image has been oriented so that the blur has occurred purely in a horizontal direction. Consider the particular kernel for which the J'th pixel in each row of the blurred image is the uniformly weighted mean of pixels

J through J+31 in the original unblurred image.

[0150] Accuracy. We implemented in the C programming language a straightforward version of the RL method that uses LPHDR arithmetic. The program reads a test image, blurs it using the kernel discussed above, then deblurs it using either fp+noise or lns arithmetic. The RL algorithm computes sums, such as when convolving the kernel with the current approximation of the deblurred image. Our implementation computes these sums using the Kahan method, discussed earlier. FIG. 7 shows the test image in original form. It is a satellite picture of a building used during Barack Obama's inauguration. FIG. 8 shows the image extremely blurred by the kernel. It is difficult to see any particular objects in this image. FIG. 9 shows the result of deblurring using standard floating point arithmetic. FIG. 10 shows the result of deblurring using fp+noise arithmetic, and FIG. 11 shows the result of deblurring using lns arithmetic. In all these cases the image is sufficiently restored that it is possible to recognize buildings, streets, parking lots, and cars.

[0151] In addition to displaying the images herein for judgement using the human eye, we computed a numerical measure of deblurring performance. We computed the mean difference, over all pixels in the image, between each original pixel value (a gray scale value from 0 to 255) and the corresponding value in the image reconstructed by the RL method. Those numerical measures are shown below in Table 1:

[0000]

TABLE 1

Image type	Mean pixel error
------------	------------------

Blurred	32.0
---------	------

RL using standard floating point	13.0
----------------------------------	------

RL using fp + noise	13.8
---------------------	------

RL using lns	14.8
--------------	------

[0152] These results, together with the subjective but important judgements made by the human eye, show that LPHDR arithmetic provides a substantial and useful degree of deblurring compared to standard floating point arithmetic. Further, in this example we chose an extreme degree of blurring, to better convey the concept and visual impact of the deblurring using LPHDR arithmetic. On more gentle and typical blur kernels, the resulting deblurred images are much closer to the originals than in this case, as can be seen by shrinking the kernel length and running the RL algorithm with LPHDR arithmetic on those more typical cases.

[0153] Efficiency. It is clear to those with ordinary skill in the art that Richardson-Lucy



using a local kernel performs only local computational operations. An image to be deblurred can be loaded into the PE array, storing one or more pixels per PE, the deconvolution operation of RL can then be iterated dozens or hundreds of times, and the deblurred image can be read back to the host processor. As long as sufficient iterations are performed, this makes efficient use of the machine.

[0154] An extreme form of image deblurring is the iterative Reconstruction method used in computational tomography. Reconstructing 3D volumes from 2D projections is an extremely computational task. The method discussed above generalizes naturally to iterative Reconstruction and makes efficient use of the machine.

[0155] Among the advantages of embodiments of the invention are one or more of the following.

[0156] PEs implemented according to certain embodiments of the present invention may be relatively small for PEs that can do arithmetic. This means that there are many PEs per unit of resource (e.g., transistor, area, volume), which in turn means that there is a large amount of arithmetic computational power per unit of resource. This enables larger problems to be solved with a given amount of resource than does traditional computer designs. For instance, a digital embodiment of the present invention built as a large silicon chip fabricated with current state of the art technology might perform tens of thousand of arithmetic operations per cycle, as opposed to hundreds in a conventional GPU or a handful in a conventional multicore CPU. These ratios reflect an architectural advantage of embodiments of the present invention that should persist as fabrication technology continues to improve, even as we reach nanotechnology or other implementations for digital and analog computing.

[0157] Doing arithmetic with few resources generally means, and in the embodiments shown specifically means, that the arithmetic is done using low power. As a result, a machine implemented in accordance with embodiments of the present invention can have extremely high performance with reasonable power (for instance in the tens of watts) or low power (for instance a fraction of a watt) with reasonably high performance. This means that such embodiments may be suitable for the full range of computing, from supercomputers, through desktops, down to mobile computing. Similarly, and since cost is generally associated with the amount of available resources, embodiments of the present invention may provide a relatively high amount of computing power per unit of cost compared to conventional computing devices.

[0158] The SIMD architecture is rather old and is frequently discarded as an approach to computer design by those having ordinary skill in the art. However, if the processing elements of a SIMD machine can be made particularly small while retaining important functionality, such as general arithmetic ability, the architecture can be useful. The embodiments presented herein have precisely those qualities.

[0159] The discovery that massive amounts of LPHDR arithmetic is useful as a fairly general computing framework, as opposed to the common belief that it is not useful, can be an advantage in any (massively or non-massively) parallel machine design or non-parallel design, not just in SIMD embodiments. It could be used in FPGAs, FPAAs, GPU/SIMT machines, MIMD machines, and in any kind of machine that uses compact arithmetic processing elements to perform large amounts of computation using a small amount of resources (like transistors or volume).

[0160] Another advantage of embodiments of the present invention is that they are not merely useful for performing computations efficiently in general, but that they can be used to tackle a variety of real-world problems which are typically assumed to require high-precision computing elements, even though such embodiments include only (or predominantly) low-precision computing elements. Although several examples of such real-world problems have been presented herein, and although we have also had success implementing non-bonded force field computations for molecular dynamics simulation and other tasks, these are merely examples and do not constitute an exhaustive set of the real-world problems that embodiments of the present invention may be used to solve.

[0161] The embodiments disclosed above are merely examples and do not constitute limitations of the present invention. Rather, embodiments of the present invention may be implemented in a variety of other ways, such as the following.

[0162] For example, embodiments of the present invention may represent values in any of a variety of ways, such as by using digital or analog representations, such as fixed point, logarithmic, or floating point representations, voltages, currents, charges, pulse width, pulse density, frequency, probability, spikes, timing, or combinations thereof. These underlying representations may be used individually or in combination to represent the LPHDR values. LPHDR arithmetic circuits may be implemented in any of a variety of ways, such as by using various digital methods (which may be parallel or serial, pipelined or not) or analog methods or combinations thereof. Arithmetic elements may be connected using various connection architectures, such as nearest 4, nearest 8, hops of varying degree, and architectures which may or may not be rectangular or grid-like. Any method may be used for communication among arithmetic elements, such as parallel or serial, digital or analog or mixed-mode communication. Arithmetic elements may operate synchronously or asynchronously, and may operate globally simultaneously or not. Arithmetic elements may be implemented, for example, on a single physical device, such as a silicon chip, or spread across multiple devices and an embodiment built from multiple devices may have its arithmetic elements connected in a variety of ways, including for example being connected as a grid, torus, hypercube, tree, or other method. Arithmetic elements may be connected to a host machine, if any, in a variety of ways, depending on the cost and bandwidth and other requirements of a particular embodiment. For example there may be many host machines connected to the collection of arithmetic elements.

[0163] Although certain embodiments of the present invention are described as being implemented as a SIMD architecture, this is merely an example and does not constitute a limitation of the present invention. For example, embodiments of the present invention may be implemented as reconfigurable architectures, such as but not limited to programmable logic devices, field programmable analog arrays, or field programmable gate array architectures, such as a design in which existing multiplier blocks of an FPGA are replaced with or supplemented by LPHDR arithmetic elements of any of the kinds disclosed herein, or for example in which LPHDR elements are included in a new or existing reconfigurable device design. As another example, embodiments of the present invention may be implemented as a GPU or SIMT-style architecture which incorporates LPHDR arithmetic elements of any of the kinds disclosed herein. For example, LPHDR elements could supplement or replace traditional arithmetic elements in current or new graphics processing unit designs. As yet another example, embodiments of the present invention may be implemented as a MIMD-style architecture which incorporates LPHDR arithmetic elements of any of the

kinds disclosed herein. For example, LPHDR arithmetic elements could supplement or replace traditional arithmetic elements in current or new MIMD computing system designs. As yet another example, embodiments of the present invention may be implemented as any kind of machine, including a massively parallel machine, which uses compact arithmetic processing elements to provide large amounts of arithmetic computing capability using a small amount of resources (for example, transistors or area or volume) compared with traditional architectures.

[0164] Although certain embodiments of the present invention are described herein as executing software, this is merely an example and does not constitute a limitation of the present invention. Alternatively, for example, embodiments of the present invention may be implemented using microcode or a hardware sequencer or state machine or other controller to control LPHDR arithmetic elements of any of the kinds disclosed herein. Alternatively, for example, embodiments of the present invention may be implemented using hardwired, burned, or otherwise pre-programmed controllers to control LPHDR arithmetic elements of any of the kinds disclosed herein.

[0165] Although certain embodiments of the present invention are described herein as being implemented using custom silicon as the hardware, this is merely an example and does not constitute a limitation of the present invention. Alternatively, for example, embodiments of the present invention may be implemented using FPGA or other reconfigurable chips as the underlying hardware, in which the FPGAs or other reconfigurable chips are configured to perform the LPHDR operations disclosed herein. As another example, embodiments of the present invention may be implemented using any programmable conventional digital or analog computing architecture (including those which use high-precision computing elements, including those which use other kinds of non-LPHDR hardware to perform LPHDR arithmetic, and including those which are massively parallel) which has been programmed with software to perform the LPHDR operations disclosed herein. For example, embodiments of the present invention may be implemented using a software emulator of the functions disclosed herein.

[0166] As yet another example, embodiments of the present invention may be implemented using 3D fabrication technologies, whether based on silicon chips or otherwise. Some example embodiments are those in which a memory chip has been bonded onto a processor or other device chip or in which several memory and/or processor or other device chips have been bonded to each other in a stack. 3D embodiments of the present invention are very useful as they may be denser than 2D embodiments and may enable 3D communication of information between the processing units, which enables more algorithms to run efficiently on those embodiments compared to 2D embodiments.

[0167] Although certain embodiments of the present invention are described herein as being implemented using silicon chip fabrication technology, this is merely an example and does not constitute a limitation of the present invention. Alternatively, for example, embodiments of the present invention may be implemented using technologies that may enable other sorts of traditional digital and analog computing processors or other devices. Examples of such technologies include various nanomechanical and nanoelectronic technologies, chemistry based technologies such as for DNA computing, nanowire and nanotube based technologies, optical technologies, mechanical technologies, biological technologies, and other technologies whether based on transistors or not that are capable of implementing LPHDR architectures of



the kinds disclosed herein.

[0168] Certain embodiments of the present invention have been described as "massively parallel" embodiments. Although certain embodiments of the present invention may include thousands, millions, or more arithmetic units, embodiments of the present invention may include any number of arithmetic units (as few as one). For example, even an embodiment which includes only a single LPHDR unit may be used within a serial processing unit or other device to provide a significant amount of LPHDR processing power in a small, inexpensive processor or other device.

[0169] For certain embodiments of the present invention, even if implemented using only digital techniques, the arithmetic operations may not yield deterministic, repeatable, or the most accurate possible results within the chosen low precision representation. For instance, on certain specific input values, an arithmetic operation may produce a result which is not the nearest value in the chosen representation to the true arithmetic result.

[0170] The degree of precision of a "low precision, high dynamic range" arithmetic element may vary from implementation to implementation. For example, in certain embodiments, a LPHDR arithmetic element produces results which include fractions, that is, values greater than zero and less than one. For example, in certain embodiments, a LPHDR arithmetic element produces results which are sometimes (or all of the time) no closer than 0.05% to the correct result (that is, the absolute value of the difference between the produced result and the correct result is no more than one-twentieth of one percent of the absolute value of the correct result). As another example, a LPHDR arithmetic element may produce results which are sometimes (or all of the time) no closer than 0.1% to the correct result. As another example, a LPHDR arithmetic element may produce results which are sometimes (or all of the time) no closer than 0.2% to the correct result. As yet another example, a LPHDR arithmetic element may produce results which are sometimes (or all of the time) no closer than 0.5% to the correct result. As yet further examples, a LPHDR arithmetic element may produce results which are sometimes (or all of the time) no closer than 1%, or 2%, or 5%, or 10%, or 20% to the correct result.

[0171] Besides having various possible degrees of precision, implementations may vary in the dynamic range of the space of values they process. For example, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range approximately from one millionth to one million. As another example, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range approximately from one billionth to one billion. As yet another example, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range approximately from one sixty five thousandth to sixty five thousand. As yet further examples, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range from any specific value between zero and one sixty five thousandth up to any specific value greater than sixty five thousand. As yet further examples, other embodiments may process values in spaces with dynamic ranges that may combine and may fall between the prior examples, for example ranging from approximately one billionth to ten million. In all of these example embodiments of the present invention, as well as in other embodiments, the values that we are discussing may be signed, so that the above descriptions characterize the absolute values of the numbers being discussed.

[0172] The frequency with which LPHDR arithmetic elements may yield only approximations to correct results may vary from implementation to implementation. For example, consider an embodiment in which LPHDR arithmetic elements can perform one or more operations (perhaps including, for example, trigonometric functions), and for each operation the LPHDR elements each accept a set of inputs drawn from a range of valid values, and for each specific set of input values the LPHDR elements each produce one or more output values (for example, simultaneously computing both sin and cos of an input), and the output values produced for a specific set of inputs may be deterministic or non-deterministic. In such an example embodiment, consider further a fraction  $F$  of the valid inputs and a relative error amount  $E$  by which the result calculated by an LPHDR element may differ from the mathematically correct result. In certain embodiments of the present invention, for each LPHDR arithmetic element, for at least one operation that the LPHDR unit is capable of performing, for at least fraction  $F$  of the possible valid inputs to that operation, for at least one output signal produced by that operation, the statistical mean, over repeated execution, of the numerical values represented by that output signal of the LPHDR unit, when executing that operation on each of those respective inputs, differs by at least  $E$  from the result of an exact mathematical calculation of the operation on those same input values, where  $F$  is 1% and  $E$  is 0.05%. In several other example embodiments,  $F$  is not 1% but instead is one of 2%, or 5%, or 10%, or 20%, or 50%. For each of these example embodiments, each with some specific value for  $F$ , there are other example embodiments in which  $E$  is not 0.05% but instead is 0.1%, or 0.2%, or 0.5%, or 1%, or 2%, or 5%, or 10%, or 20%. These varied embodiments are merely examples and do not constitute limitations of the present invention.

[0173] For certain devices (such as computers or processors or other devices) embodied according to the present invention, the number of LPHDR arithmetic elements in the device (e.g., computer or processor or other device) exceeds the number, possibly zero, of arithmetic elements in the device which are designed to perform high dynamic range arithmetic of traditional precision (that is, floating point arithmetic with a word length of 32 or more bits). If  $NL$  is the total number of LPHDR elements in such a device, and  $NH$  is the total number of elements in the device which are designed to perform high dynamic range arithmetic of traditional precision, then  $NL$  exceeds  $T(NH)$ , where  $T$  is some function. Any of a variety of functions may be used as the function  $T$ . For example, in certain embodiments,  $T(NH)$  may be twenty plus three times  $NH$ , and the number of LPHDR arithmetic elements in the device may exceed twenty more than three times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed fifty more than five times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As yet another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed one hundred more than five times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As yet another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed one thousand more than five times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As yet another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed five thousand more than five times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. Certain embodiments of the present invention may be implemented within a single physical



device, such as but not limited to a silicon chip or a chip stack or a chip package or a circuit board, and the number NL of LPHDR elements in the physical device and the number NH of elements designed to perform high dynamic range arithmetic of traditional precision in the physical device may be the total counts of the respective elements within that physical device. Certain embodiments of the present invention may be implemented in a computing system including more than one physical device, such as but not limited to a collection of silicon chips or chip stacks or chip packages or circuit boards coupled to and communicating with each other using any means (such as a bus, switch, any kind of network connection, or other means of communication), and in this case the number NL of LPHDR elements in the computing system and the number NH of elements designed to perform high dynamic range arithmetic of traditional precision in the computing system may be the total counts of the respective elements within all those physical devices jointly.

[0174] Certain embodiments of the present invention may constitute, or may be part of, processors, which are devices capable of executing software to perform computations. Such processors may include mechanisms for storing software, for using the software to determine what operations to perform, for performing those operations, for storing numerical data, for modifying data according to the software specified operations, and for communicating with devices connected to the processor. Processors may be reconfigurable devices, such as, without limitation, field programmable arrays. Processors may be co-processors to assist host machines or may be capable of operating independently of an external host. Processors may be formed as a collection of component host processors and co-processors of various types, such as CPUs, GPUs, FPGAs, or other processors or other devices, which in the art may be referred to as a heterogeneous processor design or heterogeneous computing system, some or all of which components might incorporate the same or distinct varieties of embodiments of the present invention.

[0175] Embodiments of the present invention may, however, be implemented in devices in addition to or other than processors. For example, a computer including a processor and other components (such as memory coupled to the processor by a data path), wherein the processor includes components for performing LPHDR operations in any of the ways disclosed herein, is an example of an embodiment of the present invention. More generally, any device or combination of devices, whether or not falling within the meaning of a "processor," which performs the functions disclosed herein may constitute an example of an embodiment of the present invention.

[0176] More generally, any of the techniques described above may be implemented, for example, in hardware, software tangibly stored on a computer-readable medium, firmware, or any combination thereof. The techniques described above may be implemented in one or more computer programs executing on a programmable computer including a processor, a storage medium readable by the processor (including, for example, volatile and non-volatile memory and/or storage elements), at least one input device, and at least one output device. Program code may be applied to input entered using the input device to perform the functions described and to generate output. The output may be provided to one or more output devices.

[0177] Each computer program within the scope of the claims below may be implemented in any programming language, such as assembly language, machine language, a high-level procedural programming language, or an object-oriented programming language. The programming language may, for example, be a compiled

or interpreted programming language.

[0178] Each such computer program may be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a computer processor. Method steps of the invention may be performed by a computer processor executing a program tangibly embodied on a computer-readable medium to perform functions of the invention by operating on input and generating output. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, the processor receives instructions and data from a read-only memory and/or a random access memory. Storage devices suitable for tangibly embodying computer program instructions include, for example, all forms of non-volatile memory, such as semiconductor memory devices, including EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROMs. Any of the foregoing may be supplemented by, or incorporated in, specially-designed ASICs (application-specific integrated circuits) or FPGAs (Field-Programmable Gate Arrays). A computer can generally also receive programs and data from a storage medium such as an internal disk (not shown) or a removable disk. These elements will also be found in a conventional desktop or workstation computer as well as other computers suitable for executing computer programs implementing the methods described herein, which may be used in conjunction with any digital print engine or marking engine, display monitor, or other raster output device capable of producing color or gray scale pixels on paper, film, display screen, or other output medium.

**Espacenet****Claims: JP2012530966 (A) — 2012-12-06**

---

**Processing with Compact Arithmetic Processing Element****Claims not available for JP2012530966 (A)****Claims of corresponding document: US2010325186 (A1)**

**A high quality text as facsimile in your desired language may be available amongst the following family members:**

CA2768731 (A1) KR20120040197 (A) US2010325186 (A1) WO2010148054 (A2)  
US2013031153 (A1) US2014095571 (A1)

- [Original claims](#)
- [Claims tree](#)

The EPO does not accept any responsibility for the accuracy of data and information originating from other authorities than the EPO; in particular, the EPO does not guarantee that they are complete, up-to-date or fit for specific purposes.

1. 1. A device:

comprising a plurality of low precision high dynamic range (LPHDR) execution units adapted to execute, in parallel, a first plurality of operations on a first plurality of input signals representing a first plurality of numerical values to produce a first plurality of output signals representing a second plurality of numerical values,

wherein, for each of the plurality of LPHDR execution units, for at least one operation that the LPHDR execution unit is adapted to execute, the dynamic range of the possible valid inputs to the at least one operation is at least as wide as from 1/65,000 through 65,000 and for at least 5% of the possible valid inputs to that operation, for at least one output signal produced by that operation, the statistical mean, over repeated execution on the same inputs from the at least 5% of the possible valid inputs to that operation, of the numerical values represented by the at least one output signal of the LPHDR unit executing that operation on those inputs differs by at least 0.05% from the result of an exact mathematical calculation of the operation on the numerical values of those same inputs; and

consisting of at least one high-precision execution unit adapted to execute at least one second operation on at least one second input signal representing at least one third numerical value to produce at least one second output signal representing at least one fourth numerical value,

wherein the at least one high-precision execution unit is adapted to execute at least one arithmetic operation on floating point numbers that are 32 or more bits wide;

wherein the number of LPHDR execution units in the device exceeds, by at least 20 more than three times, the number of the at least one high-precision execution unit.

2. 2. A device:

comprising a first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/65,000$  through  $65,000$  and for at least 5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least 5% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least 0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and not including any high-precision execution units adapted to execute at least one arithmetic operation on floating point numbers that are 32 or more bits wide.

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

## ⑫ 公開特許公報(A) 平3-14128

⑤ Int. Cl.<sup>5</sup>

識別記号

庁内整理番号

⑬ 公開 平成3年(1991)1月22日

G 06 F 7/00

7313-5B  
7313-5B

G 06 F 7/00

1 0 1 W  
T

審査請求 未請求 請求項の数 6 (全9頁)

⑭ 発明の名称 対数表現数値利用の演算装置

⑯ 特 願 平1-151108

⑰ 出 願 平1(1989)6月13日

⑱ 発 明 者 黒 河 富 夫 愛知県豊田市保見ヶ丘6丁目1番地1 公団110-403号  
 ⑲ 出 願 人 黒 河 富 夫 愛知県豊田市保見ヶ丘6丁目1番地1 公団110-403号

## 明 細 書

## 1. 発明の名称

対数表現数値利用の演算装置

## 2. 特許請求の範囲

1. 対数表現数値演算法利用の演算装置(2)において、固定小数点数で表される入力データを入力ルックアップテーブル(1)を使用して対数表現数値に変換し、それを対数表現数値演算法利用の演算装置(2)への入力とし、その演算結果は必要に応じて出力ルックアップテーブル(3)を使用して、固定小数点数に変換して出力する演算装置

2. 浮動小数点数を入力データとする入力ルックアップテーブル(1)とした、請求1の演算装置

3. 浮動小数点数を出力データとする出力ルックアップテーブル(3)とした、請求1の演算装置

4. 浮動小数点数を入力データとする入力ルックアップテーブル(1)とし、浮動小数点数を出

力する出力ルックアップテーブル(3)とした、請求1の演算装置

5. 対数表現数値を入力データとするが、入力ルックアップテーブル(1)を経ないで対数表現数値演算法利用の演算装置(2)へ直接入力する、請求1の演算装置

6. 対数表現数値演算法利用の演算装置(2)の出力データを、出力ルックアップテーブル(3)を経ないで出力する、請求1の演算装置

## 3. 発明の詳細な説明

(産業上の利用分野)

コンピュータなどのデジタル装置において、画像データ、音声データ、メモリのアドレス値、画像配列上の座標値などデジタル数値(特に固定小数点数)で表現されるデータを高速に精度よく演算することに関する。

(従来の技術)

従来、固定小数点数で表されたデータは、固定小数点数のまま固定小数点演算法で、または浮動小数点数に変換して浮動小数点演算法で演算され



特開平 3-14128(2)

ていた。固定小数点演算の場合演算速度は速いが、演算誤差が大きく、また、ダイナミックレンジも小さく、演算の途中結果が大きくなる場合、数値が表現できない欠点がある。浮動小数点演算で行う場合、演算誤差は比較的小さいが、演算速度が遅く、演算装置も複雑になる欠点がある。

また従来から、固定小数点演算法、浮動小数点演算法以外の演算法で、対数表現数値演算法（Logarithmic Arithmetic）という高速で精度のよい演算法が知られている（例えば、文献：Kingsbury and Rayner, "Digital filtering using logarithmic arithmetic", Electron Lett., vol.7, pp.56-58, Feb. 1971）。その数値演算の原理を以下で示す。

対数表現数値演算法について：

対数表現数値演算法においては、数値は式（1）のように表現される。

$$s d_1 d_2 \cdots d_n f_1 f_2 \cdots f_n \quad (1)$$

ここで、 $s, d_1, f_1$  は0または1で  $s$  は数値の符号、 $d$ -部と  $f$ -部の間には小数点があるとし、従って、 $d$ -

(3)

次に減算であるが、 $a^x$  から  $a^y$  を減じる場合、答えを  $a^z$  とすると、

$$a^z = a^x - a^y = a^x (1 - a^{-1} x^{-y}) \quad (x > y \text{ の場合}) \quad (5)$$

となり、対数をとると、

$$z = x + \log_a (1 - a^{-1} x^{-y}) \quad (6)$$

となる。従って、 $\log_a (1 - a^{-1} x^{-y})$  をルックアップテーブルで予め用意しておけば、 $z$  は高速に演算できる。 $x < y$  の場合は  $x$  と  $y$  を入れ替えた形で上記と同様にできる。 $x = y$  のときは真の値はゼロであるが、式（2）はゼロを表現できないのでゼロに最も近い数を与える（ルックアップテーブルでそのように用意しておく）。従って、 $z$  は負の数で表現可能な絶対値の最も大きい値となる。

累乗演算も高速にできる。例えば、ある数  $x$  の  $u$  乗（ $x^u, u$  は実数）の計算が非常に高速にできる。対数表現数値では指数を  $u$  倍すればよい。極端な例は平方と平方根の計算である。平方の計算は指数を1ビット左シフトするだけである。平方根の場合は指数を1ビット右シフトするだけである。

(5)

部は整数部、 $f$ -部は小数部である。対数の底は暗黙の値で1より大きい数  $a$  である。式（1）は式（2）の値を表す。

$$\pm a^{d d_1 d_2 \cdots d_n f_1 f_2 \cdots f_n} \quad (2)$$

従って、 $d$ -部と  $f$ -部は小数部のある固定小数点数で数値の指数である。対数表現数値演算法においては、乗除算は指数の固定小数点演算の加算または減算である。従って、演算誤差はなく、高速にできる。加算を説明する。二つの数  $a^x$  と  $a^y$  の加算を考える。その和を  $a^z$  とすると、 $z$  を求めることが加算である。

$$a^z = a^x + a^y = a^x (1 + a^{y-x}) \quad (3)$$

であるから、 $a$  を低とする対数をとると、式（3）は

$$z = x + \log_a (1 + a^{-1} y^{-x}) \quad (x > y \text{ の場合}) \quad (4)$$

となる。 $\log_a (1 + a^{-1} y^{-x})$  は  $|y-x|$  の関数であるから、 $|y-x|$  をアドレスとするルックアップテーブルにより高速に演算できる。従って、対数表現数値演算法における加算は高速にできる。 $x \leq y$  の場合は  $x$  と  $y$  を交換するだけで、同じように演算できる。

(4)

また、対数表現数値演算法は同条件（同じ程度のダイナミックレンジと一語のビット数が同じ）の浮動小数点演算よりも演算精度が高いことが知られている（例えば、文献：Kurokawa, Payne and Lee, "Error analysis of recursive digital filters implemented with logarithmic number systems", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-28, pp.706-715, Dec. 1980）。

従来、対数表現数値演算法による処理システムは第2図で示す構成になっている。対数型アナログアンプ（4）は、アナログ信号をアナログのまま対数に変換する（ $x \rightarrow \log_a x$ ）装置である。A/Dコンバータ（5）は通常のA/Dコンバータであり、アナログ信号をデジタル信号に変換する。対数表現数値演算法利用の演算装置（2）は対数表現数値演算法による演算装置である。D/Aコンバータ（6）は通常のD/Aコンバータであり、デジタル信号をアナログ信号に変換する。指数型アナログアンプ（7）はアナログ信号をアナログのまま

(6)

特開平 3-14128(3)

指数変換する ( $x \rightarrow a^x$ )。

以上のごとく構成されるシステムでは、対数型アナログアンプ(4)や指数型アナログアンプ(7)という特別のハードウェアが必要であり、しかもその特性が対数表現数値演算法利用の演算装置(2)と整合していなければならない。また、既存の画像、音声、その他データ入力装置によるデータは、そのままでは対数表現数値演算法利用の演算装置(2)への入力データとはならない。しかも、対数表現数値演算法利用の演算装置(2)はデジタル・フィルタ等の限られた分野にしか使われていない。

(発明が解決しようとする課題)

本発明は、上記従来法の欠点を解消するもので、演算速度が速くて、演算誤差の小さい、しかも簡便な演算装置を提供することを目的とする。

(問題を解決するための手段)

第1図は本発明の構成を示す図である。入力データ(固定小数点データなど)を入力ルックアップテーブル(1)を使用して、対数表現数値に変換する(7)

数が存在しているからである。例えば  $m=4$ ,  $n=10$ ,  $a=2$  の場合を考える。これは16ビットの語長で丁度可能である。この場合第3図で説明すると、横軸に  $i$  があるが、 $i=1$  から  $i$  までの間には  $2^{i-1}$  個の数があり ( $n=10$  であるから)、従って縦軸においては、 $2^{i-1}$  から  $2^i$  までの間には  $2^{i-1}$  個の数がある。一番数の密度が低いところは、255が  $N$  の上限の場合、254と255の間であるが、ここでも、その間には約6個の数がある。99から100の間には約15個の数が存在する。20ビットの語長にして、 $m=4$ ,  $n=14$ ,  $a=2$  とすると、整数254と255の間には約92個の数が存在する。従って、あまり大きくない数  $N$  であれば、相当精度よく対数表現数値に変換することが可能である。この変換はルックアップテーブルを使用すると高速にできる。すなわち、これが第1図の入力ルックアップテーブル(1)の働きである。

なお、上記の数  $N$  は整数である必要はなく、小数部のある固定小数点数あるいは浮動小数点数で

(9)

換して(入力データが対数表現数値の場合は入力ルックアップテーブル(1)を経ないで)、対数表現数値演算法利用の演算装置(2)へ入力し、そこで演算を行い、その演算結果は必要に応じて出力ルックアップテーブル(3)を使用して、固定小数点数または浮動小数点数に変換することにより、または変換しないで対数表現数値のまま出力することにより、演算を高速高精度に行うことができる。

(作用)

第3図は対数表現数値の指数  $i$  は整数ではなく、式(1)の形式の固定小数点数とそれによる  $a^i$  値の関係 ( $N = a^i$ ,  $a=2$ ) を示す図である。限られたビット数(例えば8ビット)で表されるような整数  $N$  (例えば  $N=255$ ) は、式(1)での  $m+n$  を比較的大きくとれば(例えば、 $m+n=14$  で、一語16ビット)底  $a$  を適正に決めることにより相当に精度よく式(1)の形の対数表現数値に変換できる。すなわち  $N$  は  $a^i$  の形で近似できる。これは整数  $N$  の近傍には多くの  $a^i$  形式の

(8)

もよい。いずれにしても、その数が表現されているビットパターンをアドレスとすれば、ルックアップテーブルを使用することにより、その数に応じた値を提供できる。

変換後の計算は対数表現数値演算法利用の演算装置(2)で行うのでこれも高速高精度にできる。

演算の結果を固定小数点数または浮動小数点数の形に直して出力する場合、出力の語長を必要に応じて長くすれば、精度は落ちない。固定小数点数の出力の場合も精度を要する場合、小数部のビット数を多くすればよい。これも出力ルックアップテーブル(3)で用意しておけばよい。従って、変換は高速にできる。

(発明の実施例)

実施例1

画像の幾何学的変換を第4、5図等を使用して説明する。第4図は画像の幾何学的変換を示す図である。すなわち、原画像(4.1)から幾何学的変換によって変換後の画像(4.2)を得た図である。式(7)は拡大、縮小、回転、平行移動等の

(10)

特開平 3-14128(4)

幾何学的変換をするための座標変換式（アフィン変換の式）である。

$$\begin{aligned} X_1 &= a_{11}X_0 + a_{12}Y_0 + a_{13} \\ Y_1 &= a_{21}X_0 + a_{22}Y_0 + a_{23} \end{aligned} \quad (7)$$

式(7)では $(X_0, Y_0)$ は原画像(41)の画像平面上の画素の座標を表し、 $(X_1, Y_1)$ は変換後の画像(42)の画像平面上で、 $(X_0, Y_0)$ に相当する画素の座標である。式(7)は式(8)のように書き換えることができる。

$$\begin{aligned} X_0 &= b_{11}X_1 + b_{12}Y_1 + b_{13} \\ Y_0 &= b_{21}X_1 + b_{22}Y_1 + b_{23} \end{aligned} \quad (8)$$

式(8)は変換後の画像(42)の画像平面上の座標 $(X_1, Y_1)$ の画素は原画像(41)の画像平面上のどこかの画素が相当しているかを表す。通常、 $(X_1, Y_1)$ を整数座標として、 $(X_0, Y_0)$ が丁度の整数座標になることは殆どなく、計算される座標は小数部分を含んだ形になり、原画像(41)の画素に丁度対応することはない。第5図はその様子を示す図である。計算式による座標点 $(X_0, Y_0)$ (51)は格子点(52)(整数座標)に一致することは

(11)

る。原画像(41)の画像平面上のその座標から変換後の画像(42)の画像平面上の座標 $(X_1, Y_1)$ に画素を移動する。上記のことを座標 $(X_1, Y_1)$ について繰り返すことにより、アフィン変換が高速高精度にできる。

#### 実施例2

実施例1で示したようなアドレスの計算ではなくデータ自身の演算にも本発明は有効である。既存の画像処理機器は整数データを多く扱う。少なくとも、画像データの入力またはディスプレイ装置等への出力は整数(もう少し広く固定小数点数)の形で行う場合が多い。それは通常0-255の整数である。従って、実施例1で示した演算と同様に、その演算は高速高精度に行うことができる。式(9)は2次元データ $g(m, n)$ のフーリエ変換式である。

$$G(k, l) = \sum_{m=0}^{N-1} \sum_{n=0}^{K-1} g(m, n) W_1^{mk} W_2^{nl} \quad (9)$$

ただし、 $W_1 = \exp(-j2\pi/N)$ ,  $W_2 = \exp(-j2\pi/K)$ の複素数で、 $N, K$ はそれぞれ画像 $g(m, n)$ の横と縦の画素数

(13)

殆どない。変換後の画像の画素(濃度)を適正なものにする一つの方法として、最近傍法という方法が知られている。その方法では座標点 $(X_0, Y_0)$ (51)に最も近い格子点(52)の画素を選んで変換後の座標 $(X_1, Y_1)$ の画素とする。計算は座標値 $(X_0, Y_0)$ を四捨五入によって整数座標とすることによって行う。

この応用例に対する、本発明の適用について説明する。先ず、式(8)の係数 $(b_{ij})$ を本発明で使用する対数表現数値、すなわち式(1)の形に変換する。これは通常1回だけであるから高速である必要はない。次に変換後の座標 $(X_1, Y_1)$ (整数)を入力ルックアップテーブル(1)を使用して、対数表現数値に変換し(テーブル変換だから高速)、式(8)を対数表現数値演算法利用の演算装置(2)で計算する(高速高精度)。結果は指数に相当する部分が固定小数点数であるのでそれをアドレスとする出力ルックアップテーブル(3)により四捨五入して、整数値に変換する(高速)。これがすなわち最近傍法の座標アドレス計算である

(12)

である。これについても、 $W_1, W_2$ のそれぞれを対数表現数値で与え、 $g(m, n)$ はあまり大きくない整数(例えば0-255)だとすると、実施例1と同じように高速高精度に演算できる。この演算は通常高速フーリエ変換で行うので、それが更に高速になる。ただし、出力は出力ルックアップテーブル(3)により必要なら、整数値ではなく実数値(固定小数点数、浮動小数点数または、対数表現数値)の形で出力する。

#### 実施例3

1次元のデータ、例えば、音声データ(入力ビット数8ではなく、もっと大きい数、12-14ビットで扱うことが望ましい)についても当然、高速高精度に演算できる。音声データなどの1次元データに対するフーリエ変換は式(10)により定義される。

$$G(k) = \sum_{n=0}^{N-1} g(n) W^{nk} \quad (10)$$

ただし、 $W = \exp(-j2\pi/N)$ で $N$ はデータの数である。これについても、実施例2と同じ理由で高速高精

(14)

特開平 3-14128(5)

度に演算できる。

#### 実施例 4

本発明はコンピュータグラフィックスにおける図形描画においても、非常に効果的である。式(11)は中心が原点にあり、半径がRの円の式である。

$$x^2 + y^2 = R^2 \quad (11)$$

式(11)は式(12)の様に書き換えることができる。

$$y = \pm \sqrt{R^2 - x^2} \quad (12)$$

式(12)でyが負でない部分だけをとると式(12)は式(13)となる。

$$y = \sqrt{R^2 - x^2} \quad (13)$$

式(13)において、xの0から $R/\sqrt{2}$ ( $x=y$ )までの整数値に対して、yを計算し四捨五入して整数値まで求めると、それが第6図で示すような8分円の描画点(61)である。後は、 $y=x$ の直線、x軸、y軸に対して対象な点をもとめて行けば、完全な円になる。式(13)の計算ではxの

(15)

ところである(1点プロットする毎に必ず)。

(78)は $x=y$ であるかどうかを判定するところである。(79)は $x=y$ の場合、点(x,y)をプロットするところである。

しかし、本発明によれば第6図のような8分円の描画アルゴリズムは第8図のようになる。非常に簡単である。

第8図について説明する。変数x、yは整数であり、 $r_1, x_1, y_1$ は対数表現数値、すなわち、式(1)の形式の変数を表す。Rは半径で実数である。第8図において、(81)は初期化を行うところである。半径Rを四捨五入して $RD(R)$ はRを四捨五入して整数にすることを意味する)、yとし、xをゼロ、 $r_1$ を $\log_2(R^2)$ とする( $r_1$ は値としては、ほぼ $R^2$ を表す)。(82)は終了かどうかを判定するところである。(83)は点(x,y)をプロット(描画)するところである。(84)はx座標を1つ進めるところである。(85)は整数xをルックアップテーブルT(入力ルックアップテーブル(1))により対数表現数値 $x_1$ に変換するところ

(17)

平方、減算、更に $R^2 - x^2$ の平方根を計算しなければならない。従って、通常、浮動小数点演算で計算する。しかしながら、浮動小数点演算では平方(または乗算)や平方根の計算には大変な時間がかかる。実際には、第7図で示すような増分法と言われる8分円発生アルゴリズム等が開発されていて、整数の計算だけでできる様になっている。

第7図について説明する。この図に出てくる変数は全て整数である。第7図において、(71)は初期化をするところである。xをゼロ、yを半径R、dを $3-2R$ とする。(72)は終了かどうかを判定するところである。(73)は点(x,y)をプロット(描画)するところである。(74)は次のプロット点が右横または右斜め下のどちらであるかを判定するところである。(75)は次のプロット点が右横の場合の処理で、そのように判別変数dを更新するところである。(76)は次のプロット点が右斜め下の場合の処理で、判別変数dをそのように更新し、y座標を一つ進める(71)ところである。(77)はx座標を1つ進め

(16)

るである。(86)は $\sqrt{R^2 - x^2}$ の計算を対数表現数値演算法利用の演算装置(2)で行うところである。ここでは $R^2$ は既に計算されていて、xの平方と、減算と、 $R^2 - x^2$ の平方根の計算を行う。xの平方については、 $x_1$ の指数部を2倍するか、1ビット左シフトするだけである。減算は式(6)の方法でルックアップテーブルを使って行う。 $R^2 - x^2$ の平方根の計算は上記の減算の結果(対数表現数値)を2で割るか1ビット右シフトするだけである。全て高速にできる。結果を $y_1$ とする。(87)は対数表現数値から整数値へ変換するところである。これはルックアップテーブルも(出力ルックアップテーブル(3))により四捨五入して行う。これも高速である。(88)は $x=y$ であるかどうかを判定するところである。(89)は $x=y$ の場合、点(x,y)をプロットするところである。

以上により第8図のアルゴリズムは第7図のアルゴリズムに比べ高速であることが分かる。

なお、第7図で示すアルゴリズムでは半径Rは整数( $>0$ )でなければならないが、第8図のア

(18)

特開平 3-14128(6)

ルゴリズムでは半径Rは整数である必要はない。

その点においても、本発明の方が優れている。

以上個別の実施例を示したが、これらは、単なる個別の実施例であって、高速高精度の演算が要求されるところでは、広範囲に本発明の実施が有効である。また、本発明はハードウェアで有効に実施できるが、特に集積回路での実施が特に有効である。

#### (発明の効果)

本発明により、既存の画像、音声、その他のデータ入力装置によるデータや、メモリアドレス、データ配列の座標値等、固定小数点型のデータで表わされる数値に対して、高速高精度演算ができるので、特別の対数型アナログアンプ(4)や指数型アナログアンプ(7)が必要なく、既存のコンピュータ等のデジタル装置に本発明の演算装置を加えることにより処理を高速高精度にすることができる。

請求範囲2、3、4、5及び6の入出力が固定小数点以外についての有効性について追加説明す

(19)

る。

入力または出力が浮動小数点数の場合、既存の浮動小数点型のデジタルシグナルプロセッサ等とのやり取りが高速簡便にできる。また、入力または出力が対数表現数値の場合は他の対数表現数値演算法利用の演算装置とのやり取りが高速簡便になる効果がある。

#### 4. 図面の簡単な説明

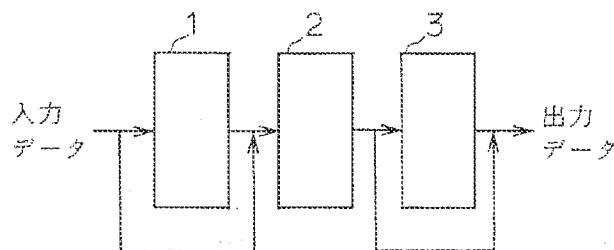
第1図は本発明の構成図、第2図は対数表現数値演算法による従来の処理システムの構成図、第3図は対数表現数値の指数iとその表現値 $a^i$ の関係図、第4図は画像の幾何学的変換の概念図、第5図は画像の幾何学的変換の計算座標 $(x_a, y_a)$ と格子点の関係を示す図、第6図は8分円の描画点の図、第7図は増分法と言われる従来の8分円の描画アルゴリズムの図、第8図は本発明を適用した8分円の描画アルゴリズムの図。

特許出願人

黒河富夫

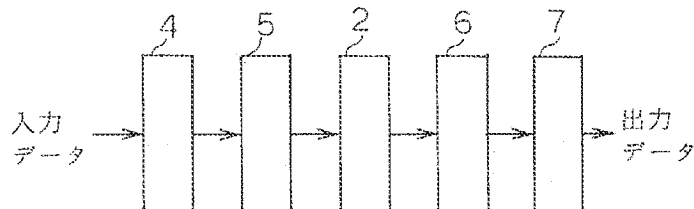
(20)

第1図



- 1 入力ルックアップテーブル  
2 対数表現数値演算法利用の演算装置  
3 出力ルックアップテーブル

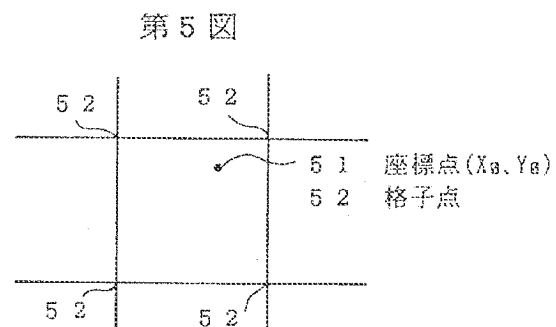
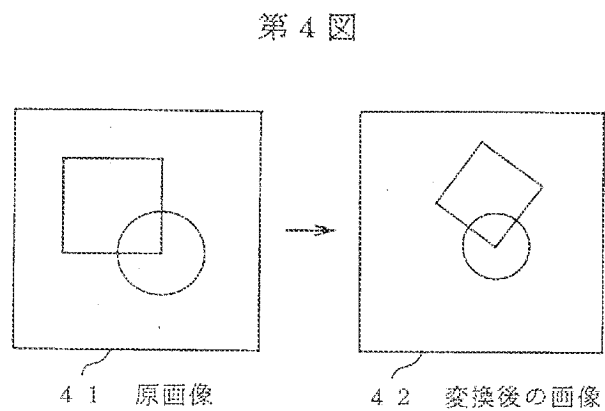
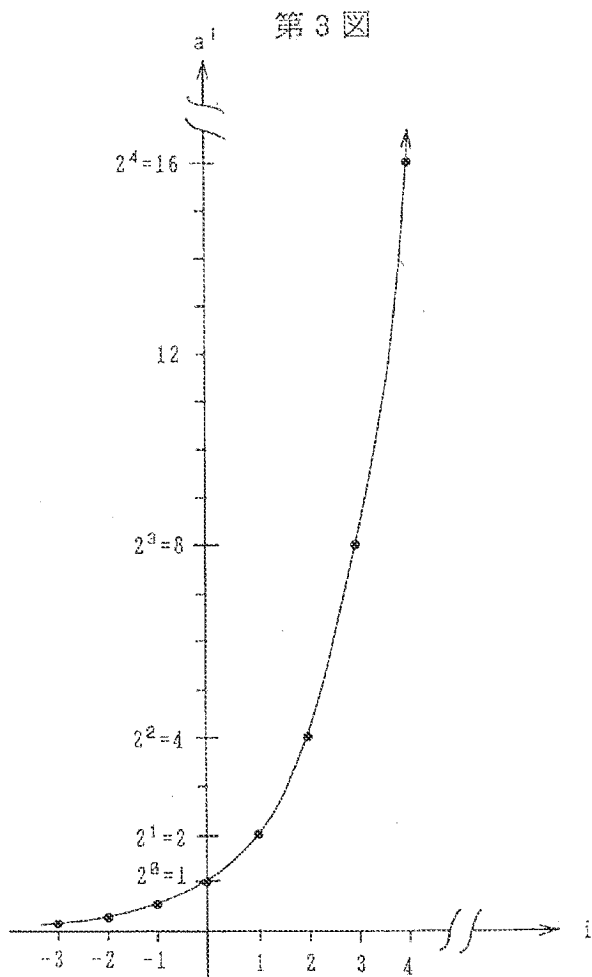
第2図



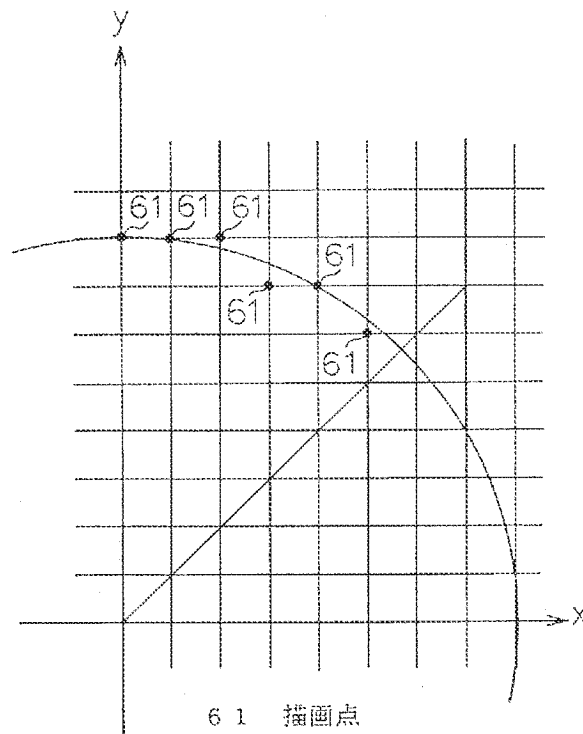
- 4 対数型アナログアンプ 5 ADコンバータ  
6 DAコンバータ 7 指数型アナログアンプ



特開平 3-14128(7)

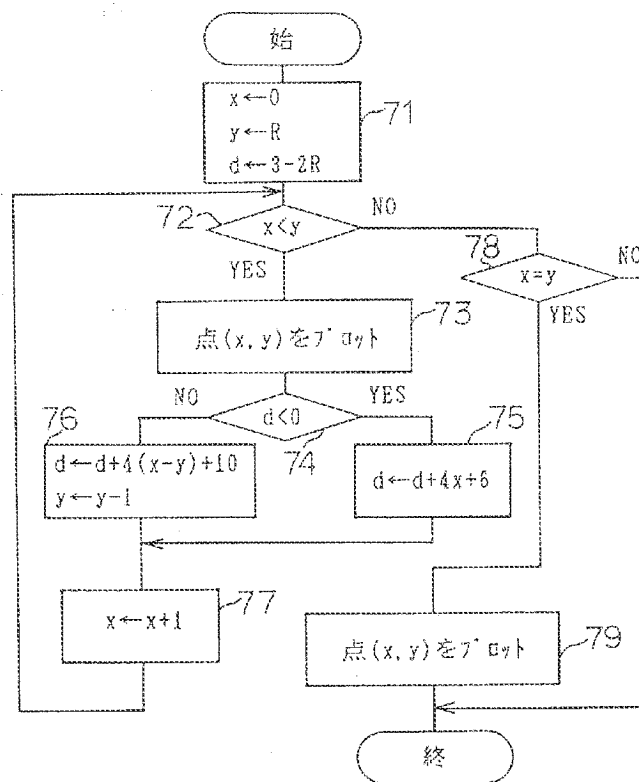


第 6 図

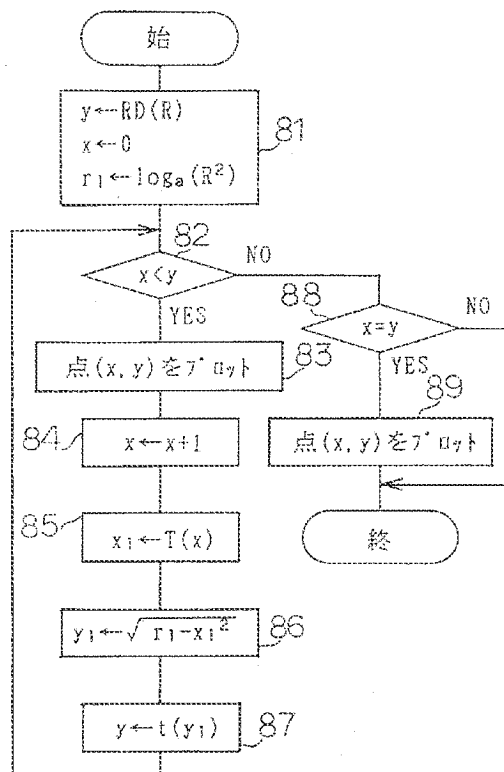


6.1 描画点

第 7 図



第 8 図





Espacenet

Bibliographic data: JPH0314128 (A) — 1991-01-22

## ARITHMETIC UNIT UTILIZING LOGARITHMIC EXPRESSION NUMERICAL VALUE

Inventor(s): KUROKAWA TOMIO ± (KUROKAWA TOMIO)

Applicant(s): KUROKAWA TOMIO ± (KUROKAWA TOMIO)

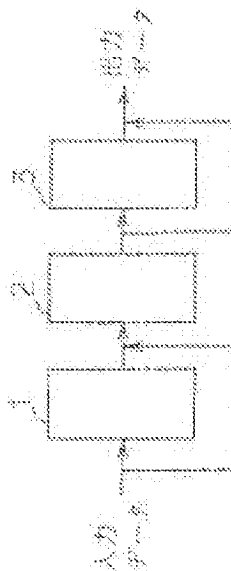
Classification: - international: **G06F7/00; G06F7/76**; (IPC1-7): G06F7/00  
 - cooperative:

Application number: JP19890151108 19890613

Priority number(s): JP19890151108 19890613

## Abstract of JPH0314128 (A)

PURPOSE: To increase arithmetic speed, and also, to decrease an arithmetic error by converting input data to a logarithmic expression numerical value by using an input look-up table, inputting it to an arithmetic unit, converting the result of its operation to the number of fixed points by using an output look-up table if necessary and outputting it. CONSTITUTION: Input data (fixed point data, etc.) is converted to a logarithmic expression numerical value by using an input look-up table 1, and inputted to an arithmetic unit 2 utilizing a logarithmic expression numerical value arithmetic method. Subsequently, an operation is executed by the arithmetic unit 2, and the result of its operation is converted to the number of fixed points or the number of floating points by using an output look-up table 3 if necessary. In such a manner, a special logarithmic analog amplifier and an exponential analog amplifier are not required, and by adding this arithmetic unit to the digital device of an existing computer, etc., the processing at high speed and with high accuracy is realized.



**Electronic Acknowledgement Receipt**

<b>EFS ID:</b>	22738522
<b>Application Number:</b>	13849606
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	6059
<b>Title of Invention:</b>	Processing with Compact Arithmetic Processing Element
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates
<b>Customer Number:</b>	24208
<b>Filer:</b>	Robert Plotkin/Karen Del Greco
<b>Filer Authorized By:</b>	Robert Plotkin
<b>Attorney Docket Number:</b>	A0006-1001C2
<b>Receipt Date:</b>	25-JUN-2015
<b>Filing Date:</b>	25-MAR-2013
<b>Time Stamp:</b>	13:07:56
<b>Application Type:</b>	Utility under 35 USC 111(a)

**Payment information:**

Submitted with Payment	no
------------------------	----

**File Listing:**

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Form (SB08)	rplotkincom-A0006-1001C2_IDS_1449.pdf	62596 9e5e39949bc3329bfa8d037bf0adcd640f41391c	no	2

**Warnings:****Information:**



This is not an USPTO supplied IDS fillable form

2	Foreign Reference	rplotkincom-A0006-1001C2-CITE-1.pdf	9317104	no	38
			9f0b369e79f310fb8119bb2aec4c814edc7f53f1		
Warnings:					
Information:					
3	Foreign Reference	rplotkincom-A0006-1001C2-CITE-2.pdf	5109821	no	79
			4cfa530cf7f593452739f30580d12ca9b5f4ce83		
Warnings:					
Information:					
4	Foreign Reference	rplotkincom-A0006-1001C2-CITE-3.pdf	554598	no	10
			10840fd9c186797f5da7e10dfb72e1bc3df0c		
Warnings:					
Information:					
5	Non Patent Literature	rplotkincom-A0006-1001C2-CITE-4.pdf	709127	no	11
			e37b684c6596d87e2aaf6df6ff8c1d82b442079		
Warnings:					
Information:					
Total Files Size (in bytes):			15753246		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

#### **New Applications Under 35 U.S.C. 111**

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

#### **National Stage of an International Application under 35 U.S.C. 371**

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

#### **New International Application Filed with the USPTO as a Receiving Office**

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.



## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
**United States Patent and Trademark Office**  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/849,606	03/25/2013	Joseph Bates	A0006-1001C2	6059

24208	7590	01/26/2015
Robert Plotkin, P.C. 15 New England Executive Park Burlington, MA 01803		

EXAMINER	
YAARY, MICHAEL D	

ART UNIT	PAPER NUMBER
2193	

NOTIFICATION DATE	DELIVERY MODE
01/26/2015	ELECTRONIC

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

RPLOTKIN@RPLOTKIN.COM

**Office Action Summary**Application No.  
13/849,606Applicant(s)  
BATES, JOSEPHExaminer  
MICHAEL D. YAARYArt Unit  
2193AIA (First Inventor to File)  
Status  
No**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --****Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 07/30/2013.  
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on \_\_\_\_\_.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on \_\_\_\_\_; the restriction requirement and election have been incorporated into this action.
- 4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims\***

- 5) ☒ Claim(s) 1-42 is/are pending in the application.  
 5a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 6) ☒ Claim(s) 9-15 and 26-32 is/are allowed.
- 7) ☒ Claim(s) 1, 16-18 and 33 is/are rejected.
- 8) ☒ Claim(s) 2-8, 19-25 and 35-42 is/are objected to.
- 9) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

\* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see [http://www.uspto.gov/patents/init\\_events/pph/index.jsp](http://www.uspto.gov/patents/init_events/pph/index.jsp) or send an inquiry to [PPHfeedback@uspto.gov](mailto:PPHfeedback@uspto.gov).

**Application Papers**

- 10) ☐ The specification is objected to by the Examiner.
- 11) ☒ The drawing(s) filed on 03/25/2013 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

**Certified copies:**

- a) ☐ All b) ☐ Some\*\* c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\*\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)  
 Paper No(s)/Mail Date 08/24/2013
- 3) ☐ Interview Summary (PTO-413)  
 Paper No(s)/Mail Date. \_\_\_\_\_
- 4) ☐ Other: \_\_\_\_\_

Application/Control Number: 13/849,606  
Art Unit: 2193

Page 2

## DETAILED ACTION

1. Claims 1-42 are pending in the application.

### ***Double Patenting***

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory double patenting rejection is appropriate where the claims at issue are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the reference application or patent either is shown to

Application/Control Number: 13/849,606

Page 3

Art Unit: 2193

be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement. A terminal disclaimer must be signed in compliance with 37 CFR 1.321(b).

The USPTO internet Web site contains terminal disclaimer forms which may be used. Please visit <http://www.uspto.gov/forms/>. The filing date of the application will determine what form should be used. A web-based eTerminal Disclaimer may be filled out completely online using web-screens. An eTerminal Disclaimer that meets all requirements is auto-processed and approved immediately upon submission. For more information about eTerminal Disclaimers, refer to <http://www.uspto.gov/patents/process/file/efs/guidance/eTD-info-I.jsp>.

Claims 1, 16-18, and 33 are rejected on the ground of nonstatutory double patenting as being unpatentable over claims 1, 36, and 33 of U.S. Patent No. 8,407,273. Although the claims at issue are not identical, they are not patentably distinct from each other because the claims are directed to the same limitations. The only differences between the claims are minor wording differences.

Application 13/849,606	US Patent 8,407,273
Claim 1 - A device comprising: at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first	Claim 1 - A device: comprising at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first



Application/Control Number: 13/849,606  
 Art Unit: 2193

Page 4

<p>numerical value to produce a first output signal representing a second numerical value,</p> <p>wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from <math>1/65,000</math> through <math>65,000</math> and for at least <math>X=5\%</math> of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least <math>X\%</math> of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least <math>Y=.05\%</math> from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; <b>and</b></p> <p><b>at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit.</b></p>	<p>numerical value to produce a first output signal representing a second numerical value,</p> <p>wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from <math>1/65,000</math> through <math>65,000</math> and for at least <math>X=5\%</math> of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least <math>X\%</math> of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least <math>Y=.05\%</math> from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.</p>
<p>Claims-16-18 - A device comprising:</p>	<p>Claim 36 - A device:</p>

Application/Control Number: 13/849,606  
Art Unit: 2193

Page 5

<p><b>a plurality of components comprising:</b></p> <p>at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,</p> <p>wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/65,000 through 65,000 and for at least <math>X=5\%</math> of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least <math>X\%</math> of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least <math>Y=.05\%</math> from the result of an exact mathematical calculation of the first operation on the</p>	<p>comprising at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,</p> <p>wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/65,000 through 65,000 and for at least <math>X=5\%</math> of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least <math>X\%</math> of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least <math>Y=.05\%</math> from the result of an exact mathematical calculation of the first</p>
--	--

Application/Control Number: 13/849,606  
Art Unit: 2193

Page 6

<p>numerical values of that same input;</p> <p><b>the plurality of components are arranged in a stack;</b></p> <p>the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.</p>	<p>operation on the numerical values of that same input;</p> <p>wherein the number of LPHDR execution units in the device exceeds the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.</p>
<p>Claim 33 - A device comprising a computer processor and a computer-readable memory storing computer program instructions, wherein the computer program instructions are executable by the processor to emulate a second device comprising:</p> <p><b>a plurality of components comprising:</b></p> <p>at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first</p>	<p>Claim 33- A device comprising a computer processor and a computer-readable memory storing computer program instructions, wherein the computer program instructions are executable by the processor to emulate a second device comprising:</p> <p>at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output</p>

Application/Control Number: 13/849,606  
Art Unit: 2193

Page 7

<p>numerical value to produce a first output signal representing a second numerical value;</p> <p>wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from <math>1/65,000</math> through <math>65,000</math> and for at least <math>X=5\%</math> of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least <math>X\%</math> of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least <math>Y=.05\%</math> from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.</p>	<p>signal representing a second numerical value;</p> <p>wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from <math>1/65,000</math> through <math>65,000</math> and for at least <math>X=5\%</math> of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least <math>X\%</math> of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least <math>Y=.05\%</math> from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.</p>
--	--

Application/Control Number: 13/849,606  
Art Unit: 2193

Page 8

***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MICHAEL D. YAARY whose telephone number is (571)270-1249. The examiner can normally be reached on Mon-Fri 9 a.m.-5:30 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chat Do can be reached on 571-272-3721. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/MICHAEL D YAARY/  
Primary Examiner, Art Unit 2193



INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application Number	13/849,606
	Filing Date	3/25/2013
	First Named Inventor	Bates
	Art Unit	2193
	Client Number	A0006
Page 1 of 2	Matter Number	1001C2

FOREIGN PATENT DOCUMENTS					
Initials	Cite No.	Document Number	Publication Date	Country	T
/M.Y./	1	H06075986	03/18/1994	JP	

Examiner Signature	/Michael Yaary/	Date Considered	01/20/2015
Examiner: Please initial if citation considered, whether or not citation is in conformance with MPEP Section 609. Please draw a line through the citation if it is not in conformance and it is not considered. Please include a copy of this form with the next communication to the applicant.			

<b>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</b>	<b>Application Number</b>	13/849,606
	<b>Filing Date</b>	3/25/2013
	<b>First Named Inventor</b>	Bates
	<b>Art Unit</b>	2193
	<b>Client Number</b>	A0006
Page 2 of 2	<b>Matter Number</b>	1001C2

<b>GENERAL</b>
<p>Pursuant to 37 C.F.R. 1.97 and 1.98 and to the duty of disclosure set forth in 37 C.F.R. 1.56, the Examiner in charge of the above-identified application is requested to consider and make of record the references listed herewith. A copy of each listed reference, other than U.S. patents/applications and references cited in a parent application, is enclosed.</p> <p>Although the information submitted herewith may be "material" to the Examiner's consideration of the subject application, this submission is not intended to constitute an admission that such information is "prior art" as to the claimed invention.</p> <p>In accordance with 37 C.F.R. 1.97(g), the filing of this Information Disclosure Statement shall not be construed to mean that a search has been made.</p>

<b>TIMING</b>
<p>In accordance with 37 CFR 1.97(b), this Information Disclosure Statement is being filed within three months of the filing of a national application other than a continued prosecution application under 37 CFR 1.53(d); within three months of the date of entry of the national stage as set forth in 37 CFR 1.491 in an international application; before the mailing of a first Office Action on the merits; or before the mailing of a first Office Action after the filing of a request for continued examination under 37 CFR 1.114.</p>

<b>CERTIFICATION STATEMENT</b>
<p>No certification statement is required. This Information Disclosure Statement is being filed in accordance with 37 CFR 1.97(b).</p>

<b>FEE</b>
<p>No fee is required. This Information Disclosure Statement is being filed in accordance with 37 CFR 1.97(b).</p> <p>If necessary, the Director is hereby authorized to charge or credit Deposit Account No. 50-1797 for any additional fees, or any underpayment or credit for overpayment in connection herewith. Please reference attorney docket number A0006-1001C2 for any such charge or credit.</p>

<b>SIGNATURE</b>			
<b>Signature</b>	/Robert Plotkin, Reg#43861/	<b>Date</b>	8/23/2013
<b>Name</b>	Robert Plotkin, Esq.	<b>Registration Number</b>	43861

## EAST Search History

## EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	892	(low near5 precision) and (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2015/01/20 12:12
L2	79	(low near5 precision) near10 (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2015/01/20 12:12
L4	79	(low near5 precision) near10 (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2015/01/20 12:13
L5	18	G06F7/\$.cpc. and (low near5 precision) near10 (high near5 dynamic)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2015/01/20 12:24
L6	73	G06F7/\$.cpc. and range and wide and mean and blur	US-PGPUB; USPAT	OR	ON	2015/01/20 13:02
S1	1	lphdr	US-PGPUB; USPAT	ADJ	ON	2012/01/24 13:37
S2	20	((("5887160") or ("5867683") or ("5809320") or ("5293500") or ("5226166") or ("4380046") or ("5170484") or ("5801715") or ("5966528") or ("6173388") or ("6405185") or ("6728871") or ("6859869") or ("7243333") or ("20050076187") or ("20060031659") or ("20070124565") or ("20070150698") or ("20070247189") or ("20080059764")).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/01/24 13:48
S3	20	((("5887160") or ("5867683") or ("5809320") or ("5293500") or ("5226166") or ("4380046") or ("5170484") or ("5801715") or ("5966528") or ("6173388") or ("6405185") or ("6728871") or ("6859869") or ("7243333") or ("20050076187") or ("20060031659") or ("20070124565") or ("20070150698") or ("20070247189") or ("20080059764")).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/01/24 15:27
S4	13	S3 and (width or wide)	US-PGPUB; USPAT	OR	ON	2012/01/24 15:27
S5	20	((("5887160") or ("5867683") or ("5809320") or ("5293500") or ("5226166") or ("4380046") or ("5170484") or ("5801715") or ("5966528") or ("6173388") or ("6405185") or ("6728871") or ("6859869") or ("7243333") or ("20050076187") or ("20060031659") or ("20070124565") or ("20070150698") or	US-PGPUB; USPAT; USOCR	OR	OFF	2012/02/06 10:38

		("20070247189") or ("20080059764").PN.				
S6	6	S5 and precision	US-PGPUB; USPAT	OR	ON	2012/02/06 10:42
S7	1	("20100325186").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/02/06 10:46
S8	13	S5 and (width or wide)	US-PGPUB; USPAT	OR	ON	2012/02/06 11:02
S9	6	S8 and precision	US-PGPUB; USPAT	OR	ON	2012/02/06 11:02
S10	22628	(low near5 precision)	US-PGPUB; USPAT	OR	ON	2012/02/06 14:10
S11	632	(low near5 precision) and (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2012/02/06 14:11
S12	59	(low near5 precision) near10 (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2012/02/06 14:11
S13	46	(low near5 precision) near10 (high near5 dynamic) and signal\$1	US-PGPUB; USPAT	OR	ON	2012/02/06 14:11
S14	4323	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution)	US-PGPUB; USPAT	OR	ON	2012/02/06 15:04
S15	611	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution) and mean	US-PGPUB; USPAT	OR	ON	2012/02/06 15:05
S16	140	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution) and mean and precision	US-PGPUB; USPAT	OR	ON	2012/02/06 15:05
S17	28	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution) and mean and ((high or low) near5 precision)	US-PGPUB; USPAT	OR	ON	2012/02/06 15:05
S18	1	("20080276232").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/02/06 15:31
S19	1	("7549145").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/02/06 15:34
S20	682	382/255.ccls.	US-PGPUB; USPAT	OR	ON	2012/02/07 10:34
S21	494	382/255.ccls. and range	US-PGPUB; USPAT	OR	ON	2012/02/07 10:35
S22	208	382/255.ccls. and range and wide	US-PGPUB; USPAT	OR	ON	2012/02/07 10:35
S23	86	382/255.ccls. and range and wide and mean	US-PGPUB; USPAT	OR	ON	2012/02/07 10:35
S24	42	382/255.ccls. and range and wide and mean and blur	US-PGPUB; USPAT	OR	ON	2012/02/07 10:35
S25	92	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm. and mean	US-PGPUB; USPAT	OR	ON	2012/02/07 10:37
S26	1	("20080276232").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/02/07 10:50
S35	1	("8150902").PN.	USPAT; USOCR	OR	OFF	2012/11/14 09:41
S36	696	(low near5 precision) and (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2012/11/15 10:38



S37	63	(low near5 precision) near10 (high near5 dynamic)	US-PGPUB; USPAT	OR	ON	2012/11/15 10:39
S38	2	(low near5 precision) near10 (high near5 dynamic) and fpga	US-PGPUB; USPAT	OR	ON	2012/11/15 10:39
S39	2	(low near5 precision) near10 (high near5 dynamic) and "708".clas.	US-PGPUB; USPAT	OR	ON	2012/11/15 10:39
S40	20	((("5887160") or ("5867683") or ("5809320") or ("5293500") or ("5226166") or ("4380046") or ("5170484") or ("5801715") or ("5966528") or ("6173388") or ("6405185") or ("6728871") or ("6859869") or ("7243333") or ("20050076187") or ("20060031659") or ("20070124565") or ("20070150698") or ("20070247189") or ("20080059764"))).PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2012/11/15 10:40
S43	1	("8407273").PN.	USPAT; USOCR	OR	OFF	2015/01/20 10:40

**EAST Search History (Interference)**

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S27	60	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm. and mean	USPAT; UPAD	OR	ON	2012/02/07 10:37
S28	8	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm. and mean.clm.	USPAT; UPAD	OR	ON	2012/02/07 10:37
S29	0	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm. and mean.clm. and precision.clm.	USPAT; UPAD	OR	ON	2012/02/07 10:38
S30	6	("708".clas. or "712".clas. or "382.clas") and mean.clm. and precision.clm.	USPAT; UPAD	OR	ON	2012/02/07 10:38
S31	17	("708".clas. or "712".clas. or "382.clas") and (low near3 precision).clm.	USPAT; UPAD	OR	ON	2012/02/07 10:39
S32	0	("708".clas. or "712".clas. or "382.clas") and (low near3 precision).clm. and deblur.clm.	USPAT; UPAD	OR	ON	2012/02/07 10:39
S33	12	("708".clas. or "712".clas. or "382.clas") and (low near3 precision).clm. and (high near3 precision)	USPAT; UPAD	OR	ON	2012/02/07 10:40
S34	8	("708".clas. or "712".clas. or "382.clas") and (low near3 precision).clm. and (high near3 precision).clm.	USPAT; UPAD	OR	ON	2012/02/07 10:40
S41	528	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm.	USPAT; UPAD	OR	ON	2012/11/15 10:43
S42	528	("708".clas. or "712".clas. or "382.clas") and (parallel near5 execution).clm.	USPAT; UPAD	OR	ON	2012/11/15 10:43

**1/ 20/ 2015 1:13:24 PM****C:\Users\myaary\Documents\EAST Workspaces\12816201.wsp**





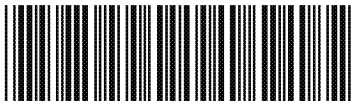
## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
 United States Patent and Trademark Office  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

## BIB DATA SHEET

CONFIRMATION NO. 6059

SERIAL NUMBER	FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.		
13/849,606	03/25/2013	708	2193	A0006-1001C2		
<b>RULE</b>						
<b>APPLICANTS</b> Singular Computing LLC, Newton, MA, Assignee (with 37 CFR 1.172 Interest); <b>INVENTORS</b> Joseph Bates, Newton, MA; <b>** CONTINUING DATA *****</b> This application is a CON of 13/399,884 02/17/2012 PAT 8407273 which is a CON of 12/816,201 06/15/2010 PAT 8150902 which claims benefit of 61/218,691 06/19/2009 <b>** FOREIGN APPLICATIONS *****</b> <b>** IF REQUIRED, FOREIGN FILING LICENSE GRANTED *** SMALL ENTITY **</b>						
Foreign Priority claimed <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No 35 USC 119(a-d) conditions met <input type="checkbox"/> Yes <input type="checkbox"/> No Verified and Acknowledged <u>/MICHAEL D YAARY/</u> Examiner's Signature		<input type="checkbox"/> Met after Allowance Initials	<b>STATE OR COUNTRY</b>  MA	<b>SHEETS DRAWINGS</b>  11	<b>TOTAL CLAIMS</b>  42	<b>INDEPENDENT CLAIMS</b>  5
<b>ADDRESS</b> Robert Plotkin, P.C. 15 New England Executive Park Burlington, MA 01803 UNITED STATES						
<b>TITLE</b> Processing with Compact Arithmetic Processing Element						
<b>FILING FEE RECEIVED</b> 2100	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:			<input type="checkbox"/> All Fees <input type="checkbox"/> 1.16 Fees (Filing) <input type="checkbox"/> 1.17 Fees (Processing Ext. of time) <input type="checkbox"/> 1.18 Fees (Issue) <input type="checkbox"/> Other _____ <input type="checkbox"/> Credit		

<b>Search Notes</b> 	<b>Application/Control No.</b> 13849606	<b>Applicant(s)/Patent Under Reexamination</b> BATES, JOSEPH
	<b>Examiner</b> MICHAEL D YAARY	<b>Art Unit</b> 2193

CPC- SEARCHED		
Symbol	Date	Examiner

CPC COMBINATION SETS - SEARCHED		
Symbol	Date	Examiner

US CLASSIFICATION SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES		
Search Notes	Date	Examiner
EAST search with keywords	01/20/2014	MY
NPL search	01/20/2014	MY
CPC search	01/20/2014	MY
Inventor name search	01/20/2014	MY

INTERFERENCE SEARCH			
US Class/ CPC Symbol	US Subclass / CPC Group	Date	Examiner

--	--

<b><i>Index of Claims</i></b>  	<b>Application/Control No.</b>  13849606	<b>Applicant(s)/Patent Under Reexamination</b>  BATES, JOSEPH
	<b>Examiner</b>  MICHAEL D YAARY	<b>Art Unit</b>  2193

✓	<b>Rejected</b>	-	<b>Cancelled</b>	N	<b>Non-Elected</b>	A	<b>Appeal</b>
=	<b>Allowed</b>	÷	<b>Restricted</b>	I	<b>Interference</b>	O	<b>Objected</b>

<input type="checkbox"/> Claims renumbered in the same order as presented by applicant			<input type="checkbox"/> CPA			<input type="checkbox"/> T.D.			<input type="checkbox"/> R.1.47		
CLAIM		DATE									
Final	Original	01/20/2015									
	1	✓									
	2	○									
	3	○									
	4	○									
	5	○									
	6	○									
	7	○									
	8	○									
	9	=									
	10	=									
	11	=									
	12	=									
	13	=									
	14	=									
	15	=									
	16	✓									
	17	✓									
	18	✓									
	19	○									
	20	○									
	21	○									
	22	○									
	23	○									
	24	○									
	25	○									
	26	=									
	27	=									
	28	=									
	29	=									
	30	=									
	31	=									
	32	=									
	33	✓									
	34	○									
	35	○									
	36	○									

<b><i>Index of Claims</i></b>  	<b>Application/Control No.</b>  13849606	<b>Applicant(s)/Patent Under Reexamination</b>  BATES, JOSEPH
	<b>Examiner</b>  MICHAEL D YAARY	<b>Art Unit</b>  2193

✓	<b>Rejected</b>	-	<b>Cancelled</b>	N	<b>Non-Elected</b>	A	<b>Appeal</b>
=	<b>Allowed</b>	÷	<b>Restricted</b>	I	<b>Interference</b>	O	<b>Objected</b>

<input type="checkbox"/> Claims renumbered in the same order as presented by applicant		<input type="checkbox"/> CPA		<input type="checkbox"/> T.D.		<input type="checkbox"/> R.1.47			
CLAIM		DATE							
Final	Original	01/20/2015							
	37	○							
	38	○							
	39	○							
	40	○							
	41	○							
	42	○							



## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
 United States Patent and Trademark Office  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/849,606	03/25/2013	Joseph Bates	A0006-1001C2

CONFIRMATION NO. 6059

## PUBLICATION NOTICE

24208  
 Robert Plotkin, P.C.  
 15 New England Executive Park  
 Burlington, MA 01803



\*OC000000067548414\*

**Title:**Processing with Compact Arithmetic Processing Element

**Publication No.**US-2014-0095571-A1

**Publication Date:**04/03/2014

## NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at [www.uspto.gov](http://www.uspto.gov). The direct link to access the publication is currently <http://www.uspto.gov/patft/>.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Office of Public Records. The Office of Public Records can be reached by telephone at (703) 308-9726 or (800) 972-6382, by facsimile at (703) 305-8759, by mail addressed to the United States Patent and Trademark Office, Office of Public Records, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at [www.uspto.gov](http://www.uspto.gov) using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently <http://pair.uspto.gov/>. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101



UNITED STATES PATENT AND TRADEMARK OFFICE  
COMMISSIONER FOR PATENTS  
P.O.BOX 1450  
ALEXANDRIA VA 22313-1451

PRESORTED  
FIRST-CLASS MAIL  
U.S. POSTAGE PAID  
POSTEDIGITAL  
NNNNN

ROBERT PLOTKIN, PC  
15 New England Executive Office Park  
Burlington, MA 01803



**Courtesy Reminder for  
Application Serial No: 13/849,606**

Attorney Docket No: A0006-1001C2

Customer Number: 24208

Date of Electronic Notification: 12/27/2013

This is a courtesy reminder that new correspondence is available for this application. If you have not done so already, please review the correspondence. The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

An email notification regarding the correspondence was sent to the following email address(es) associated with your customer number:  
RLOTKIN@RLOTKIN.COM

To view your correspondence online or update your email addresses, please visit us anytime at <https://portal.uspto.gov/secure/myportal/privatepair>. If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov or call 1-866-217-9197.



## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
 United States Patent and Trademark Office  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/849,606	03/25/2013	Joseph Bates	A0006-1001C2

CONFIRMATION NO. 6059

NEW OR REVISED PPD NOTICE

24208  
 ROBERT PLOTKIN, PC  
 15 New England Executive Office Park  
 Burlington, MA 01803



\*OC000000065756154\*

**NOTICE OF NEW OR REVISED PROJECTED PUBLICATION DATE**

The above-identified application has a new or revised projected publication date. The current projected publication date for this application is 04/03/2014. If this is a new projected publication date (there was no previous projected publication date), the application has been cleared by Licensing & Review or a secrecy order has been rescinded and the application is now in the publication queue.

If this is a revised projected publication date (one that is different from a previously communicated projected publication date), the publication date has been revised due to processing delays in the USPTO or the abandonment and subsequent revival of an application. The application is anticipated to be published on a date that is more than six weeks different from the originally-projected publication date.

More detailed publication information is available through the private side of Patent Application Information Retrieval (PAIR) System. The direct link to access PAIR is currently <http://pair.uspto.gov>. Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Questions relating to this Notice should be directed to the Office of Data Management, Application Assistance Unit at (571) 272-4000, or (571) 272-4200, or 1-888-786-0101.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT	Application Number	13/849,606
	Filing Date	3/25/2013
	First Named Inventor	Bates
	Art Unit	2193
	Client Number	A0006
Page 1 of 2	Matter Number	1001C2

FOREIGN PATENT DOCUMENTS					
Initials	Cite No.	Document Number	Publication Date	Country	T
	1	H06075986	03/18/1994	JP	

Examiner Signature		Date Considered	
Examiner: Please initial if citation considered, whether or not citation is in conformance with MPEP Section 609. Please draw a line through the citation if it is not in conformance and it is not considered. Please include a copy of this form with the next communication to the applicant.			

<b>INFORMATION DISCLOSURE STATEMENT BY APPLICANT</b>	<b>Application Number</b>	13/849,606
	<b>Filing Date</b>	3/25/2013
	<b>First Named Inventor</b>	Bates
	<b>Art Unit</b>	2193
	<b>Client Number</b>	A0006
Page 2 of 2	<b>Matter Number</b>	1001C2

<b>GENERAL</b>
<p>Pursuant to 37 C.F.R. 1.97 and 1.98 and to the duty of disclosure set forth in 37 C.F.R. 1.56, the Examiner in charge of the above-identified application is requested to consider and make of record the references listed herewith. A copy of each listed reference, other than U.S. patents/applications and references cited in a parent application, is enclosed.</p> <p>Although the information submitted herewith may be "material" to the Examiner's consideration of the subject application, this submission is not intended to constitute an admission that such information is "prior art" as to the claimed invention.</p> <p>In accordance with 37 C.F.R. 1.97(g), the filing of this Information Disclosure Statement shall not be construed to mean that a search has been made.</p>

<b>TIMING</b>
<p>In accordance with 37 CFR 1.97(b), this Information Disclosure Statement is being filed within three months of the filing of a national application other than a continued prosecution application under 37 CFR 1.53(d); within three months of the date of entry of the national stage as set forth in 37 CFR 1.491 in an international application; before the mailing of a first Office Action on the merits; or before the mailing of a first Office Action after the filing of a request for continued examination under 37 CFR 1.114.</p>

<b>CERTIFICATION STATEMENT</b>
<p>No certification statement is required. This Information Disclosure Statement is being filed in accordance with 37 CFR 1.97(b).</p>

<b>FEE</b>
<p>No fee is required. This Information Disclosure Statement is being filed in accordance with 37 CFR 1.97(b).</p> <p>If necessary, the Director is hereby authorized to charge or credit Deposit Account No. 50-1797 for any additional fees, or any underpayment or credit for overpayment in connection herewith. Please reference attorney docket number A0006-1001C2 for any such charge or credit.</p>

<b>SIGNATURE</b>			
Signature	/Robert Plotkin, Reg#43861/	Date	8/23/2013
Name	Robert Plotkin, Esq.	Registration Number	43861

(10)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-75986

(43)公開日 平成6年(1994)3月18日

(51)Int.Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/328		8320-5L		
15/16	3 9 0 T	9190-5L		

審査請求 有 請求項の数16(全 28 頁)

(21)出願番号 特願平5-50603

(22)出願日 平成5年(1993)3月11日

(31)優先権主張番号 8 6 4 1 1 2

(32)優先日 1992年4月6日

(33)優先権主張国 米国(US)

(71)出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州アーモンク (番地なし)

(72)発明者 ジェラルド・ジョージ・ベチャネック  
13760ニューヨーク州エンドウェル、デボネア・ドライブ 945

(74)代理人 弁理士 頼宮 孝一 (外4名)

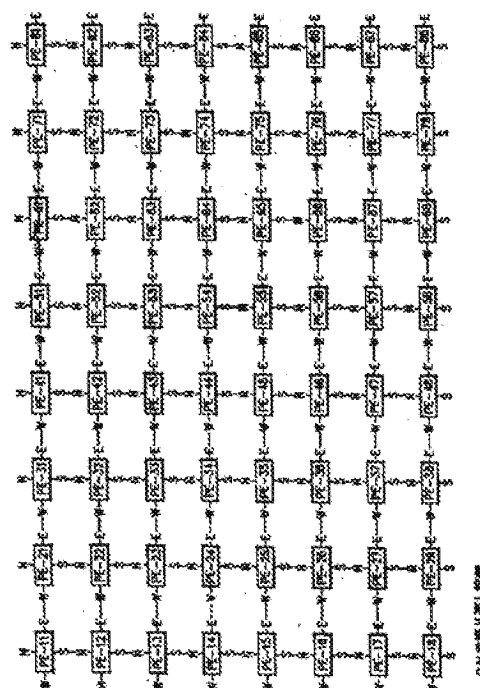
最終頁に続く

(54)【発明の名称】 大規模並列コンピューティング・システム装置

(57)【要約】

【目的】 大規模並列コンピューティング・システムに用いることのできるコンピュータ・システムの新規なPE及び関連した編成の提供。

【構成】 最近隣メッシュ・コンピュータは $N \times N$ 方形アレイのプロセッサ要素(PE)からなり、各PEが東西南北のPEのみに接続されている。PEの間に単線のインタフェースを想定した場合、メッシュ構造内には合計 $2N'$ 本のワイヤが存在する。たとえば、すべてのPEが北へデータの転送を行うSIMD操作を想定した場合、対称的な処理要素を重ね合わせ、北-南のワイヤと東-西のワイヤを共用し、これによって、配線を $N'$ 本に半減することによって、アレイを再構成することが可能である。これによって、マトリックス変形操作を2重対称処理要素内のデータ要素を交換するだけで1サイクルで行えるようになる。





(2)

特開平6-75986

1

2

## 【特許請求の範囲】

【請求項1】微分方程式を解くために有限差分法を使用することを含んでおり、処理要素ユニットのプロセッサ・アレイからなり、該プロセッサ・アレイが単一及び2重プロセッサを含んでおり、これらが命令及びデータ記憶装置を含んでおり、命令及びデータを受け取り、命令を実行し、さらにプロセッサ相互接続構成と、データ初期設定、並列機能、折り返し、及び同報プロセッサ通信をサポートする手段とからなっているマルチメディア及び汎用用途の大規模並列コンピューティング・システム装置。

【請求項2】対角線に沿って折り畳まれ、折り返し相互接続構成によって最近隣のものによって相互接続されている単一プロセッサ対角ユニット及び2重プロセッサ汎用ユニットによって構成されている $N \times N$ のマトリックスの形態で配置された $N^2$ 個の処理装置をさらに含んでいる請求項1記載の装置。

【請求項3】各プロセッサを元の $N \times N$ のマトリックスに対する参照表記によって識別してから、折り返しで、 $N \times N$ のマトリックス・アルゴリズムの三角形アレイ・アルゴリズムへの移送をサポートする請求項2記載の装置。

【請求項4】各々が折り返し通信によって最近隣に対して北、南、東及び西の入出力ポートを処理する、対角線に沿って折り畳まれ、北及び南の入出力ポートを東及び西の入出力ポートと共用することを可能とする $N \times N$ のマトリックスの形に配置された $N^2$ 個の処理装置からなっている請求項2記載の装置。

【請求項5】プロセッサ間で単一方向通信方式を利用する非競合プロセッサ間通信の機構を備えた $N \times N$ のマトリックス・アルゴリズムをアレイ・プロセッサに利用できる請求項4記載の装置。

【請求項6】各前記処理装置が折り畳み前に、元の $N \times N$ のマトリックスに対する2つの添字表記 $P_{ij}$ によって識別される請求項3記載の識別表記。

【請求項7】 $K$ がプロセッサ間のワイヤの数であり、ビット・シリアル・インタフェースの場合に $K$ が1 ( $K=1$ )となり、 $K(N^2)$ の相互接続線がある請求項2記載の装置。

【請求項8】 $PE_{ij}$ という前記単一プロセッサ対角ユニットがデータ記憶要素と、実行ユニットと、命令及びデータの通信のための同報インタフェースと、初期設定をサポートするデータ記憶インタフェースと、プロセッサ間インタフェースと呼ぶ折り返しインタフェースを備えた最近隣と、通信手段とからなっている請求項2記載の装置。

【請求項9】各々がデータ記憶要素と、実行ユニットと、命令及びデータの通信のための同報インタフェースと、初期設定をサポートするデータ記憶インタフェース

とからなっている $PE_{ij}$ 及び $PE_{ji}$ という2重プロセッサ汎用ユニットが、折り返しインタフェースを備えた最近隣に対して2重個別内部プロセッサをサポートする少なくとも1つの共通要素と、プロセッサ間インタフェースと呼ぶ通信手段とからなっている請求項2記載の装置。

【請求項10】論理演算機能を実行し、少なくとも $N \times N$ の方形アレイの処理要素( $PE$ )との外部通信を可能とする手段からなるコンピュータ機能アレイであって、複数の $PE$ がアレイの一部であり、複数の処理要素の各々が北、南、東及び西の $PE$ に対する接続のための手段を有しており、コンピュータ・インタフェースを $PE$ 間に設け、ネットワーク構造をもたらし、該ネットワーク構造が処理要素を互いに結合し、北-南のワイヤを東-西のワイヤと共用することによって、対称処理が配線をたとえばデータを北へ転送するすべての $PE$ と共用できるようにし、これによってパフォーマンスに影響を及ぼさず配線の複雑度を減少させるコンピュータ機能アレイ。

【請求項11】演算機能を実行するための手段を有している処理要素と、 $PE$ を互いに結合して、2重対称処理要素内でのデータ要素の単純な交換によって1サイクルで達成されるマトリックス変形操作をもたすための結果として対角線で折り畳まれたアレイ・プロセッサを形成し、イメージ処理及びマルチメディア用に、ならびに微分方程式を解くための差分方程式のために並列2-D畳み込み機構を提供する手段とからなるコンピュータ機能アレイ。

【請求項12】両方の内部 $PE$ に対する共通レジスタ( $W_{ij}$ )を有する対称 $PE$ 「セル」と、2つのピクセル・レジスタ( $P_{ij}$ 及び $P_{ji}$ )と、マトリックス( $P$ )またはその転置( $P^T$ )の選択を可能とするセレクタと、2つの結果レジスタ( $R_{ij}$ 及び $R_{ji}$ )と、2つの受信レジスタ( $R_{ij}$ 及び $R_{ji}$ )とからなるコンピュータ機能アレイ要素。

【請求項13】 $PE$ セルがビット・シリアル乗算器と、ビット・シリアル加算器とを含んでいる請求項12記載のコンピュータ機能アレイ要素。

【請求項14】同報コマンドを複号する単一の「コマンド制御」論理によって制御される対称2重 $PE$ 「セル」で使用される4つの入出力ポートをさらに含んでいる請求項12記載のコンピュータ機能アレイ要素。

【請求項15】処理要素間に対称2重 $PE$ セルとして折り返し接続を構成し、

1. 北発信南受信
  - ・ $PE_{ij}$ は $N/W$ ワイヤ上で $PE_{ji}$ に送信する
  - ・ $PE_{ji}$ は $S/E$ ワイヤ上で $PE_{ij}$ から受信する
  - ・ $PE_{ij}$ は $W/N$ ワイヤ上で $PE_{ji}$ に送信する
  - ・ $PE_{ji}$ は $E/S$ ワイヤ上で $PE_{ij}$ から受信する
2. 南発信北受信

(3)

特開平 6-75986

3

- ・  $PE_{ij}$  は S/Eワイヤ上で  $PE_{i,j-1}$  に送信する
- ・  $PE_{ij}$  は N/Wワイヤ上で  $PE_{i,j-1}$  から受信する
- ・  $PE_{ij}$  は E/Sワイヤ上で  $PE_{i,j-1}$  に送信する
- ・  $PE_{ij}$  は W/Nワイヤ上で  $PE_{i,j-1}$  から受信する

### 3. 東莞信西受信

- ・  $PE_n$  は E/Sワイヤ上で  $PE_{n-1}$  に送信する
- ・  $PE_n$  は W/Nワイヤ上で  $PE_{n-1}$  から受信する
- ・  $PE_n$  は S/Eワイヤ上で  $PE_{n-1}$  に送信する
- ・  $PE_n$  は N/Wワイヤ上で  $PE_{n-1}$  から受信する

#### 4. 西羌信東受信

- ・  $PE_{ij}$  は W/Nワイヤ上で  $PE_{i-1,j}$  に送信する
- ・  $PE_{ij}$  は E/Sワイヤ上で  $PE_{i,j-1}$  から受信する
- ・  $PE_{ij}$  は N/Wワイヤ上で  $PE_{i,j+1}$  に送信する
- ・  $PE_{ij}$  は S/Eワイヤ上で  $PE_{i,j+1}$  から受信する

どのようにPEセルを接続するステップからなる  
 コンピュータ機能アレイ要素によって実行される演算を  
 構成する方法。

【請求項16】PEセルのアレイが単一プロセッサ対角セルを含んでおり、 $i=j$ の場合に、対角「セル」に対する情報の転送が行われ、その転送が

### 1. 北拜信南受信

- ・PE<sub>0</sub> はW/Nワイヤ上でPE<sub>100</sub> に送信する
- ・PE<sub>0</sub> はS/Eワイヤ上でPE<sub>100</sub> から受信する

## 2. 南送信北受信

- ・PE<sub>n</sub> はS/Eワイヤ上でPE<sub>m</sub> に送信する
- ・PE<sub>n</sub> はW/Nワイヤ上でPE<sub>m</sub> から受信する

### 3. 東楚信西楚信

- ・ PE<sub>in</sub> はE/Sワイヤ上でPE<sub>out</sub> に送信する
- ・ PE<sub>in</sub> はW/Nワイヤ上でPE<sub>out</sub> から受信する

#### 4. 西羌信來受信

- ・ PE<sub>11</sub> は W/Nワイヤ上で PE<sub>12</sub> に送信する
- ・ PE<sub>11</sub> は E/Sワイヤ上で PE<sub>12</sub> から受信する

請求項15記載の演算を構成する方法。

【発明の詳細な説明】

【産業上の利用分野】本発明はコンピュータ、詳細にいえば、大規模並列アレイ・プロセッサに関する。

[0002]

【従来の技術】まず、本明細書で用いられる用語について説明する。

[0003] - ALU

ALUとはプロセッサの論理演算機構部分をいう。

[0004]・74

アレイとは1つまたは複数の次元での要素の配列をいう。アレイ・プロセッサはアレイで処理を行うように、多くの機能単位ないしPEが配列され、相互に接続されているコンピュータである。大規模並列機械はアレイ処理要素ないしアレイ要素によるデータ・アレイの並列処理のためにアレイ・プロセッサを使用する。アレイはFORTRANなどの言語では単一の名前によって識別されるデ



データ項目（アレイ要素）の順序づけられたセットを含むことができるが、他の言語では、データ項目の順序づけられたセットのこのような名前は、データ要素の順序づけられた集合ないしセットを指すものであり、これらはすべて同一の属性を有している。プログラム・アレイは数値または次元属性によって一般に指定された次元を有している。アレイの宣言子も何らかの言語のアレイの各次元のサイズを指定することができる。言語によっては、アレイはテーブル内の要素の配列となる。ハードウェアの面でいえば、アレイは大規模な並列アーキテクチャではほぼ同一のものである構造（機能エレメント）の集合である。データ並列コンピューティングにおけるアレイ要素は、演算を割り当てることができ、並列が独立しており、並列である場合に、必要な演算を実行することができる要素である。一般に、アレイは処理要素のグリッドと考えることができる。アレイのセクションにはセクション・データを割り当てることのできるので、セクション・データを規則グリッド・パターンで移動することができる。ただし、データには索引を付けることも、あるいはアレイ内の任意の位置を割り当てることもできる。

【0005】・機能單位

機能単位はある目的を達成できるハードウェア、ソフトウェアまたはこれら両方のエンティティーである。

[0006] · MIMD

アレイ内の各プロセッサがそれ自体の命令ストリームを有しているプロセッサ・アレイ・アーキテクチャであり、したがって、複数命令ストリームが処理要素当たり1つが配置されている複数データ・ストリームを処理できる。

【0007】・モジュール

モジュールは離散しており、識別可能なプログラム単位、または他のコンポーネントとともに使用するよう設計されたハードウェアの機能単位である。

100081 - PP

PEは処理要素に使用される語である。PEという用語は、割り振られたメモリ、及び本発明の並列アレイ処理要素の1つを形成する入出力可能システム要素ないし単位を相互接続した単一のプロセッサを指すのに使用される。書込みの結果として、本システムにおいては、対称的な複製可能な要素が相互接続バスを共用するために互いに接続される。

[0009] · SIMD

アレイ内のすべてのプロセッサが単一命令ストリームからのコマンドを受け、処理要素当たり1つが配置されている複数データ・ストリームを実行するプロセッサ・アレイ・アーキテクチャである。

【0010】次に、本発明の背景について述べる。視覚情報の処理は、3種類の処理ドメイン、すなわちイメージ処理、パターン認識、及びコンピュータ・グラフィッ

5

クスからなるものと考えることができる。イメージ処理、パターン認識及びコンピュータ・グラフィックスを組み合わせることをイメージ・コンピューティングと呼び、将来のマルチメディア・ワークステーションが必要とする機能を表す。「マルチメディアとは、イメージ、グラフィックス、ビデオ、オーディオ、及びテキストなどによる2つ以上の方法で情報を表示し、情報の包含性を強化し、人間とコンピュータの対話を改善する技法をいう。」

【0011】より高速なコンピュータを絶えることなく求める過程で、エンジニアたちは何百もの、場合によっては数千もの低価格のマイクロプロセッサを並列につなぎ合わせ、今日の機械の悩みの種となっている複雑な問題を解決するため分離するスーパーコンピュータを作成している。このような機械を大規模並列機械と呼ぶ。並列に作動する多重コンピュータは何10年もの間存在してきた。初期の並列機械としては、1960年代に始まったILLIACが挙げられる。他の多重プロセッサとしては(米国特許第4975834号明細書抄録参照)、シーダ、シグマ-1、パタフライ及びモナーク、インテルipsc、コネクション・マシーン、カルテックCOSMIC、Nキューブ、IBMのRP3、IBMのGF11、NYUウルトラ・コンピュータ、インテル・デルタ及びタッチストーンが挙げられる。

【0012】ILLIACから始まった大規模多重プロセッサはスーパーコンピュータと考えられる。市場で大成功を収めたスーパーコンピュータはクレイ・リサーチY-MPシステム、IBM3090、ならびにアムダール、日立、富士通及びNECを始めとする他の製造業者の機械に代表される多重ベクトル・プロセッサに基づくものである。

【0013】大規模並列プロセッサ(MPP)は今ではスーパーコンピュータになれるものと考えられている。これらのコンピュータ・システムは多数のマイクロプロセッサを、相互接続ネットワーク及びこれらのプロセッサを並列に作動させるプログラムによって集積したものである。これらのコンピュータには2つの作動モードがある。これらの機械のあるものはMIMDモード機械であり、あるものはSIMDモード機械である。これらの機械の内市場でもっとも成功を収めたものはおそらく、シンキング・マシーンス・インクのコネクション・マシーン・シリーズ1及び2である。これらは本質的にSIMD機械である。大規模並列機械の多くは並列に接続されたマイクロプロセッサを使用して、これらの並行性ないし並列作動能力を得ている。1860などのインテルのマイクロプロセッサがインテルその他によって使用されている。Nキューブはインテル386マイクロプロセッサを使用してこのような機械を作っている。その他の機械は「トランスピュータ」チップと呼ばれるものを使用して構築されている。インモス・トランスピュータI

(4)

特開平6-75986

6

MS T800はその一例である。インモス・トランスピュータT800は内蔵高速浮動小数点プロセッサを有する32ビット・デバイスである。

【0014】構築されているこの種のシステムの例としては、いくつかのインモス・トランスピュータT800チップで、各々は32の通信リンク入力と32リンクの出力を有している。各チップは単一のプロセッサ、少量のメモリ、ならびにローカル・メモリ及び外部インタフェースに対する通信リンクを有している。さらに、システムを構成するために、1MS C011及びC012などの通信リンク・アダプタが接続されている。さらに、1MS C004などのスイッチを設け、32リンクの入力と32リンクの出力の間に、たとえば、クロスバー・スイッチを設け、付加的なトランスピュータ・チップの間に2点間接続を行っている。さらに、特定の装置、グラフィックスまたはディスク・コントローラの要件に合わせて調整された特別な目的にトランスピュータを使用できるようにするトランスピュータ用の特別な回路及びインタフェース・チップがある。インモス1MS M212は16ビット・プロセスであり、オン・チップ・メモリ及び通信リンクを備えている。これはディスク駆動機構を制御するためのハードウェア及び論理回路を含んでおり、プログラム式ディスク・コントローラとして、あるいは汎用インタフェースとして使用することができる。並行性(並列操作)を使用するために、インモスはトランスピュータ用に特別な言語Occamを開発した。プログラムはトランスピュータのネットワークを直接Occamプログラムで記述する必要がある。

【0015】これらの大規模並列プロセッサの中には、異なるトポロジで相互接続されているプロセッサ・チップの並列プロセッサ・アレイを使用しているものがある。トランスピュータは1MS C004チップを追加することによって、クロスバー・ネットワークをもたらし。他のシステムのなかには、ハイパーキューブ接続を使用しているものもある。他のものはバスまたはメッシュを使用して、マイクロプロセッサを接続しており、かつ回路が関連づけられている。プロセッサ・アドレス可能ネットワークとしてスイッチを使用する回路スイッチ・プロセッサによって相互接続されているものもある。一般に、機械を互いに結合することによって、昨秋ローレンス・リバモアで相互接続された14台のRISC/6000などのように、プロセッサ・アドレス可能ネットワークは粗粒(coarse-grained)マルチプロセッサとみなされている。

【0016】いくつかの超大型機械がインテル及びNキューブ、ならびにその他の会社によって、データ処理において「大命題(グランド・チャレンジ)」と呼ばれているものを解決するために構築されている。しかしながら、これらのコンピュータはきわめて高価なものである。「大命題」を解決するため、その開発に米国政府が



(5)

特開平6-75986

7

資金を与えたコンピュータに対して最近予測されたコストは、3千万ドルから7千5百万ドルの範囲である（テラ・コンピュータ）。これらの「大命題」としては、気象モデリング、流体の乱流、汚染物の分散、人間のゲノム及び海流のマッピング、量子クロモ力学、半導体及びスーパーコンピュータのモデリング、燃焼系、視覚ならびに認識などの問題が挙げられる。

【0017】

【発明が解決しようとする課題】大規模並列アレイ・プロセッサがイメージ・コンピューティングの問題を適切に解くという問題が存在している。イメージ処理に使用されている特定のアルゴリズムの1つは畳み込みであり、これは各イメージ・ピクセル値を定義された包囲領域すなわちピクセルのウィンドウ内のピクセルの加重合計と置き換える。M×M平方畳み込みウィンドウは各々がウィンドウ内に配置されている関連したピクセルに対応しているM×Mの重みの集合からなっている。N×Nのアレイのピクセルの場合、畳み込みアルゴリズムにはM<sup>2</sup>N<sup>2</sup>の乗算が必要である。Nが1024、Mが3の単一のイメージ・フレームを想定した場合、畳み込みは畳み込み当たり9百万の乗算と積の合計計算を行い、1秒当たり30フレームの速度で発生するビデオ・データに対する処理の場合には、1秒当たり270百万の乗算と積の合計計算が必要となる。各畳み込みウィンドウの重み値を個別に取り出すことが必要であり、乗算と加算を別の演算として取り扱い、その後、加重平均ピクセル結果の書き込みを行う、このデータを処理するユニプロセッサの場合、畳み込みはピクセル当たり27回の個別な計算となり（9回の読取り、9回の乗算、8回の加算、及び1回の書き込み）、1秒当たり27百万×30回の演算すなわち1秒当たり810百万回の演算となる。計算負荷が高いため、システム・プロセッサをイメージ処理タスクから解除し、イメージ・コンピューティングに必要な適切なスループットを提供する特殊プロセッサが提案されている。このような特殊プロセッサの1つは最近隣メッシュ接続コンピュータで、複数の処理要素（PE）が東西南北の近隣PEと接続され、すべてのPEが同期単一命令複数データ（SIMD）方式で作動するものである。PEが近隣するPEのいずれとも通信できるが、一度に通信できるのは近隣するPEのうちの1つだけであると想定する。たとえば、各PEは1通信サイクルで、その東側のPEと通信できる。また、1同報通信期間に、データ及び命令をすべてのPEに同時に通信できるようにする同報機構が存在しているものと想定する。シンキング・マシーンのCM-1ファミリーに存在しているようなビット・シリアル・インタフェースが典型的なものである。

【0018】このように認識した場合、必要なものはイメージ処理を改善し、速度を上げ、大規模並列環境において並列アレイ・プロセッサの一部として複製すること

8

のできるPEである。微分方程式を解くために使用されるシステム装置を改善する必要がある。この問題には新しい種類のPEが必要と思われる。新しいPEの作成、ならびに新しい考えによって構築されたシステム装置は、マルチメディア・イメージ・コンピュータ分野において処理する必要のある複合処理を改善するものである。

【0019】

【課題を解決するための手段】本発明で行われる改善は新しい機械装置をもたらす。本発明を実施する機械をオラクル機械と呼び、これを以下で説明する。本発明は大規模並列コンピューティング・システムの作成を可能とする装置に関する。並列アレイ・コンピュータ・システムすなわち大規模並列アレイ・プロセッサに用いることのできるコンピュータ・システムの新規なPE及び関連した編成を提供する。

【0020】微分方程式を解くのに有限差分法を使用することを含む、マルチメディア用及び汎用の大規模並列コンピュータ・システムを提供する。本発明のプロセッサは三角プロセッサ・アレイ構造である。本発明のプロセッサ・アレイ構造は命令及びデータの記憶装置を含んでおり、命令とデータを受け取り、命令を実行する単一及び2重処理要素、ならびにデータの初期化、並列機能、折り返し、及び同報プロセッサ通信をサポートするためのプロセッサ相互接続編成及び方法を有している。

【0021】コンピュータは対角線に沿って折り畳まれ、折り返し相互接続構造を備えた最近隣のものによって相互接続された単一プロセッサ対角ユニットと2重プロセッサ汎用ユニットを構成しているN×Nのマトリックスの形態で配置されたN<sup>2</sup>個の処理装置を有していることが好ましい。コンピュータにおいて、各処理要素すなわちPEはマトリックスの単位である。各プロセッサを元のN×Nのマトリックスの参照符号によって識別してから、N×Nのマトリックス・アルゴリズムの三角アレイ・アルゴリズムへの移送をサポートする折り畳みを行う。

【0022】折り畳み前に、各PEは4つのポートを有しており、対角線に沿って折り畳まれており、北及び南の入出力ポートの東及び西の入出力ポートとの共有を可能とするN×Nのマトリックスの形で配置された、折り返し通信によって最近隣のものに対して北、南、東及び西の入出力ポートを各々が所有するN<sup>2</sup>個の処理装置がある。

【0023】N×Nのマトリックスを備えた本発明のプロセッサの場合、プロセッサを接続する方法は、非競合プロセッサ間通信の機構をもたらすプロセスによる。たとえば、プロセッサの間の単一方向通信方式を利用する機構を、オラクル・アレイ・プロセッサに利用する。非競合プロセッサ間通信の機構は、すべてのプロセッサに単一方向で、しかも同一方向の通信方式を利用するよう

(6)

特開平6-75986

9

要求することによって得られる。

【0024】本明細書の場合、上記の各処理要素は折り畳み前の元の $N \times N$ マトリックスに関する2つの添字を付した表記 $PE_{i,j}$ によって識別される。したがって、計算機は $K(N^2)$ の相互接続線を有することになる。ただし、 $K$ はプロセッサ間のワイヤの数であり、ビット・シリアル・インタフェースの場合、 $K$ は1になる( $K=1$ )。本発明は単一プロセッサ対角ユニットをサポートする。装置はデータ記憶要素、実行ユニット、命令及びデータの通信のための同報インタフェース、初期化をサポートするデータ記憶インタフェース、ならびにプロセッサ間インタフェースと呼ばれる折り返しのある最近隣インタフェースを含む $PE_{i,j}$ という単一プロセッサ対角ユニット、及び通信手段を有している。

【0025】本発明は以下で詳述する新規の計算機能も提供する。

【0026】これら及びその他の改善について、以下で詳細に説明する。本発明の利益及び特徴についてよりよく理解するには、以下の説明及び図面を参照されたい。

【0027】

【実施例】

<序>

【0028】図面において、メッシュ・コンピュータの例が図1に示されている。配線の複雑度を最小限のものとするために、 $PE$ 間にビット・シリアル・インタフェースを想定する。図1において、処理要素には $PE_{i,j}$ というラベルが付けられており、「 $i$ 」はマトリックスの列を示し、「 $j$ 」はマトリックスの行を示している。各処理要素 $PE_{i,j}$ は北(N)、東(E)、南(S)及び西(W)という4つのインタフェース・ポートを含んでいる。折り返し接続、すなわちトラス構成によって、各行は $N$ 本の相互接続線を含んでおり、 $N$ 個の行には、 $N^2$ 本の水平相互接続線がある。各列は $N$ 本の相互接続線を含んでおり、 $N$ 個の列には、 $N^2$ 本の垂直相互接続線がある。折り返し接続によるメッシュ接続コンピュータの線の総数は $2N^2(K)$ である。ただし、 $K$ はプロセッサ間相互接続線の数に等しく、ビット・シリアル・インタフェースの場合、 $K$ は1になる。

【0029】本発明にしたがい、本発明をビット・シリアル・インタフェースに関して説明する。なお、他の手段によって、効果的な並列インタフェースを得ることができることに留意すべきである。米国特許第07/799602号はプロトコルを使用して、この出願に記載されているマルチシステム環境でビット・シリアル通信を並列通信とインタフェースする方法を開示している。

【0030】適切なネットワーク接続によって、米国特許第07/798788号に記載されているように、MIMD、SIMD及びSISDモードの間の動的な切換えを可能とするシステムに、本発明装置を用いることができる。本発明の $PE$ は各ノードがその $PE$ 及びメモ

10

り、ならびに外部の他のノードと通信するためのポートを有している、多くのノードで構成された並列アレイ・プロセッサの一部を形成することができる。

【0031】図2はイメージ処理畳み込みアルゴリズムを実施するのに使用される典型的なメッシュ $PE$ を示しており、N、S、E及びWの送受信ポートが明確に示されている。 $PE$ は4つの内部レジスタ $W_{ij}$ 、 $P_{ij}$ 、 $R_{ij}$ 及び $R_{ij}^-$ からなっている。 $P_{ij}$ レジスタはLSSDを通るスキャン・パスであってもかまわないイメージ初期化パスによって初期化された場合に $i$ 番目のピクセル値を保持している。 $W_{ij}$ レジスタには同報機構によって畳み込みウィンドウの要素がロードされ、計算結果は $R_{ij}^-$ レジスタにロードされる。 $PE$ 間で伝送される結果は $R_{ij}$ レジスタに受け取られる。 $PE$ は「\*」で示されるビット・シリアル乗算器、及び「+」で示されるビット・シリアル加算器も含んでいる。通信を行うため、各 $PE$ は復号された同報コマンドに基づいて送信/受信方向が「コマンド制御」論理によって制御される4つの入出力ポートを含んでいる。「コマンド制御」論理によって、1つだけのポートが送信を行い、1つのポートが受信を行うことが同時にできるようになる。たとえば、ポート「N」が送信している場合には、ポート「S」が受信を行い、ポート「W」が送信している場合には、ポート「E」が受信を行う。これは $PE$ 間の情報の単一方向伝送の制約によるものである。

【0032】図1のメッシュ接続コンピュータの構成及び図2の $PE$ の内部構造において、構造内の任意の伝送の際に、 $PE$ 接続線の内50%が利用されるにすぎないことになる。接続線の半数で、メッシュ接続コンピュータと同じ機能と性能を達成することが望ましいが、これは $N^2$ 線の節減がチップ面積の対応する節減によってもたらされるからである。オラクル・コンピュータの構成はこの能力を達成する。オラクル構成の $PE$ を作成するプロセスの第1段階は、 $PE_{i,j}$ 対角要素に沿ってメッシュ接続コンピュータを折り畳む、すなわち「 $i=1$ 」にすることである。たとえば、図1に示す $8 \times 8$ のメッシュを折り畳むと、対称の $PE$ 、すなわち $PE_{i,j}$ 及び $PE_{j,i}$ が重ねられた図3及び図4に示すオラクル・メッシュになる。対称の $PE$ は図3及び図4に模式的に示すように送信及び受信ポートを共用しており、頂部 $PE$ の $PE_{i,j}$ 、W、S、N及びEポートが底部 $PE$ の $PE_{j,i}$ 、N、E、W及びSポートのそれぞれと共用されている。対称 $PE$ の入出力ポートのこの共用は、図5ないし図4のオラクル対称 $PE$ 「セル」構造に詳細に示されている。図5において、内部対称 $PE$ 「セル」は内部 $PE$ 両方に対する共通 $W_{ij}$ レジスタ、 $P_{ij}$ 及び $P_{ij}$ という2つのピクセル・レジスタ、マトリックス $P$ またはその転置 $P^T$ の選択を可能とするセレクト、2つの結果レジスタ $R_{ij}^-$ 及び $R_{ij}^+$ 、ならびに2つの受信レジスタ $R_{ij}$ 及び $R_{ij}$ からなっている。 $PE$ は「\*」で示されるビット・シリア



(7)

特開平6-75986

11

ル乗算器、及び「+」で示されるビット・シリアル加算器も含んでいる。同報コマンドを復号する単一の「コマンド制御」論理によって制御される4つの入出力ポートだけが、対称2重PE「セル」に利用される。「コマンド制御」論理は入出力ポートの方向だけでなく、入出力ポートを送信及び受信レジスタにリンクするスイッチの設定も制御する。共用入出力ポートは（頂部PE方向ポート・ラベル）／（底部PE方向ポート・ラベル）で示されるが、ただし、PE<sub>11</sub>は頂部PEを表し、PE<sub>12</sub>は底部PEを表す。頂部及び底部PEの表記の選択は、検討及び表示を容易とするためだけのものである。

【0033】図6ないし図9は4つの伝送モードをサポートする内部スイッチの設定を示す。図6において、「コマンド制御」はスイッチ及び送信機／受信機を、南（S）から受信して、北（N）へ送信するように設定している。頂部及び底部両方のPEは北へ送信するとともに、両方とも南から情報を受信する。これは図6の矢印の次の頂部PE<sub>11</sub>の送信に対する「N／」という表記及び底部PE<sub>12</sub>の送信に対する「／N」という表記によって示されている。受信した情報は図6の矢印、ならびに頂部PE<sub>11</sub>の受信ポートに対する「S／」という表記及び底部PE<sub>12</sub>の受信ポートに対する「／S」という表記によって示されている。対称PE「セル」の4つの入出力ポートに対するこの表記を使用して、簡単な記号体系を図15に示すように対称PEについて構成することができるが、この図において、PE<sub>11</sub>は頂部PEであり、PE<sub>12</sub>は底部PEである。オラクル構成にこの記号体系を利用すると、セル間の配線の規則性を示す簡単な図16ないし図17が得られる。

【0034】2重プロセッサの内部スイッチは次の8つの接続点A、B、C、D、ならびにW、X、Y及びZからなっている。

- ・点Aはプロセッサ間インタフェースからデータを受信するプロセッサP<sub>11</sub>のレジスタR<sub>11</sub>に接続されている。
- ・点Bはプロセッサ間インタフェースにデータを供給するプロセッサP<sub>11</sub>のレジスタR<sub>11</sub>に接続されている。
- ・点Cはプロセッサ間インタフェースにデータを供給するプロセッサP<sub>12</sub>のレジスタR<sub>12</sub>に接続されている。
- ・点Dはプロセッサ間インタフェースからデータを受信するプロセッサP<sub>12</sub>のレジスタR<sub>12</sub>に接続されている。
- ・点WはPE<sub>11</sub>の西及びPE<sub>12</sub>の北の近隣PEの間でのデータの送信及び受信のための受信／送信機構に接続されている。
- ・点XはPE<sub>11</sub>の南及びPE<sub>12</sub>の東の近隣PEの間でのデータの送信及び受信のための受信／送信機構に接続されている。

12

・点YはPE<sub>11</sub>の北及びPE<sub>12</sub>の西の近隣PEの間でのデータの送信及び受信のための受信／送信機構に接続されている。

・点ZはPE<sub>11</sub>の東及びPE<sub>12</sub>の南の近隣PEの間でのデータの送信及び受信のための受信／送信機構に接続されている。

【0035】2重プロセッサ・スイッチはスイッチの状態に応じて、点A、B、C、Dならびに点W、X、Y及びZの間に接続／非接続パスをもたらす。スイッチの1つの状態では、点AとW、BとZ、CとX、及びDとYの間の接続パスが、東発信西受信に対するものとなる。スイッチの第2の状態では、点AとX、BとY、CとW、及びDとZの間の接続パスが、北発信南受信に対するものとなる。スイッチの第3の状態では、点AとY、BとX、CとZ、及びDとWの間の接続パスが、南発信北受信に対するものとなり、スイッチの第4の状態では、点AとZ、BとW、CとY、及びDとXの間の接続パスが、西発信東受信に対するものとなる。受信／発信機構が、接続されている受信PEに対してPEからの信号を駆動する1つの状態に各々が応じ、かつ接続されている発信PEからPEへの信号を受信する他の状態に応じる4つの双方向ドライバ／レシーバ機構からなっており、かつ4つの双方向ドライバ／レシーバ機構の内、東発信西受信、北発信南受信、南発信北受信、及び西発信東受信という4つのケースに関して、2つだけが同時にデータを送信し、2つが同時にデータを受信するようにする制御が与えられることに留意すべきである。

【0036】対角PE（図10）は西／北ポート及び南／東ポートを共用しており、対角PE「セル」当たり2つだけのポートを必要とする。図11ないし図14は4つの伝送モードをサポートする内部スイッチの設定を示す。対角プロセッサの内部スイッチ機構は4つの接続点A、B、X及びYからなっており、点Aはデータ受信レジスタR<sub>11</sub>に接続されており、点Bはデータをプロセッサ間インタフェースに供給するレジスタR<sub>12</sub>に接続されており、点Xは対角PEの西及び北（W／N）の近隣PEの間でのデータの送信及び受信のために受信／送信機構に接続されており、点Yは対角PEの南及び東（S／E）の近隣PEの間でのデータの送信及び受信のために受信／送信機構に接続されている。対角スイッチはスイッチの状態に応じて、点A、Bと点X、Yの間に接続／非接続パスをもたらす。スイッチの1つの状態において、点AとXの間、及び点BとYの間の接続パスが2つの送信／受信ケース、すなわち点Yを通る南送信、点Xを通る北受信、ならびに点Yを通る東送信、点Xを通る西受信に対してもたらされる。スイッチの第2の状態において、点AとYの間、及び点BとXの間の接続パスが2つの送信／受信ケース、すなわち点Xを通る北送信、点Yを通る南受信、ならびに点Xを通る西送信、点Yを通る東受信に対してもたらされる。受信／送信機構が1

(8)

特開平6-75986

13

つの状態に応じて各々がPEから接続されている受信PEへの信号を駆動し、かつもう1つの状態に応じて各々が接続されている送信PEからPEへの信号を受信する2つの双方向ドライバ/レシーバ機構からなっており、双方向ドライバ/レシーバ機構が両方とも同時にデータを駆動したり、あるいは両方とも同時にデータを受信したりすることがないようにするための制御が設けられていることに留意すべきである。

【0037】方形メッシュ構造を対角線に沿って折り畳\*

14

\*むと、方形メッシュの頂縁が西側の縁部に重なり、方形メッシュの東側縁部が南側縁部と重なる。方形アレイの北/南側縁部と東/西側縁部との間にある方形メッシュ・コンピュータ構成の2N折り返し接続を、N折り返し接続のみを必要とするオラクル・メッシュ構成で共用することができる。オラクル水平相互接続線を計数し、折り返し接続を水平ワイヤとみなすと、次のようになる。

【数1】

$$\text{オラクル水平及び折り返しワイヤ} = (1+2+\cdots+N)K = \left(N \frac{(N+1)}{2}\right)K$$

【0038】オラクル垂直相互接続線を計数すると次のようになる ※【数2】

$$\text{オラクル・ワイヤの総数} = \left(N \frac{(N+1)}{2} + N \frac{(N-1)}{2}\right)K = N^2(K)$$

【0039】オラクル・ワイヤの総数は次のようになる。 ★【数3】

$$\text{オラクル・ワイヤの総数} = \left(N \frac{(N+1)}{2} + N \frac{(N-1)}{2}\right)K = N^2(K)$$

【0040】ただし、Kはプロセッサ間相互接続線の数であり、ビット・シリアル・インタフェースの場合、Kは1である。オラクル・ワイヤの総数が方形メッシュ構成で必要とされるワイヤの数のちょうど半分であることが実証された。

【0041】次いで、PE間の単一方向情報転送の制限のもとで、2つのコンピュータ構成が機能的に同等であることが実証された。次の4つの場合を検討する必要がある。

1. 北発信南受信
2. 南発信北受信
3. 東発信西受信
4. 西発信東受信

【0042】PE間で伝送される情報の宛先点がオラクルにおいて、メッシュ接続構成の場合と同じであることが示されている。PE<sub>ij</sub>の場合、次のようになる。

1. 北発信PE<sub>ij</sub> → PE<sub>i,j+1</sub>
2. 南発信PE<sub>ij</sub> → PE<sub>i,j-1</sub>
3. 東発信PE<sub>ij</sub> → PE<sub>i+1,j</sub>
4. 西発信PE<sub>ij</sub> → PE<sub>i-1,j</sub>

【0043】ただし、i-1=0またはj-1=0の場合には、折り返し接続に対してi=Nまたはj=Nにセットし、i+1>Nまたはj+1>Nの場合には、i=1またはj=1にセットする。

【0044】オラクルにおいて、まず、「i」と「j」の値が交換されるのであるから、頂部PE「セル」ではi<j、底部PE「セル」ではi>jである対称2重PE「セ

ル」を考察する。この最初の考察は次に考察する対角セル以外のすべてのPEを説明するものである。対称2重PE「セル」について、i-1=0またはj-1=0の場合には、折り返し接続に対してi=Nまたはj=Nにセットし、i+1>Nまたはj+1>Nの場合には、i=1またはj=1にセットする。

【0045】1. 北発信南受信

- ・PE<sub>ij</sub>はN/Wワイヤ上でPE<sub>i,j+1</sub>に送信する
- ・PE<sub>ij</sub>はS/Eワイヤ上でPE<sub>i,j-1</sub>から受信する
- ・PE<sub>ij</sub>はW/Nワイヤ上でPE<sub>i+1,j</sub>に送信する
- ・PE<sub>ij</sub>はE/Sワイヤ上でPE<sub>i-1,j</sub>から受信する

【0046】2. 南発信北受信

- ・PE<sub>ij</sub>はS/Eワイヤ上でPE<sub>i,j+1</sub>に送信する
- ・PE<sub>ij</sub>はN/Wワイヤ上でPE<sub>i,j-1</sub>から受信する
- ・PE<sub>ij</sub>はE/Sワイヤ上でPE<sub>i+1,j</sub>に送信する
- ・PE<sub>ij</sub>はW/Nワイヤ上でPE<sub>i-1,j</sub>から受信する

【0047】3. 東発信西受信

- ・PE<sub>ij</sub>はE/Sワイヤ上でPE<sub>i+1,j</sub>に送信する
- ・PE<sub>ij</sub>はW/Nワイヤ上でPE<sub>i-1,j</sub>から受信する
- ・PE<sub>ij</sub>はS/Eワイヤ上でPE<sub>i,j+1</sub>に送信する
- ・PE<sub>ij</sub>はN/Wワイヤ上でPE<sub>i,j-1</sub>から受信する

【0048】4. 西発信東受信

- ・PE<sub>ij</sub>はW/Nワイヤ上でPE<sub>i+1,j</sub>に送信する
- ・PE<sub>ij</sub>はE/Sワイヤ上でPE<sub>i-1,j</sub>から受信する
- ・PE<sub>ij</sub>はN/Wワイヤ上でPE<sub>i,j+1</sub>に送信する
- ・PE<sub>ij</sub>はS/Eワイヤ上でPE<sub>i,j-1</sub>から受信する

【0049】対角「セル」について、i=jの場合、次

15

の情報の転送が生じる。

【0050】1. 北発信南受信

- ・  $PE_{ij}$  は  $W/N$ ワイヤ上で  $PE_{i,j+1}$  に送信する
- ・  $PE_{ij}$  は  $S/E$ ワイヤ上で  $PE_{i+1,j}$  から受信する

【0051】2. 南発信北受信

- ・  $PE_{ij}$  は  $S/E$ ワイヤ上で  $PE_{i,j+1}$  に送信する
- ・  $PE_{ij}$  は  $W/N$ ワイヤ上で  $PE_{i+1,j}$  から受信する

【0052】3. 東発信西受信

- ・  $PE_{ij}$  は  $E/S$ ワイヤ上で  $PE_{i,j+1}$  に送信する
- ・  $PE_{ij}$  は  $W/N$ ワイヤ上で  $PE_{i+1,j}$  から受信する

【0053】4. 西発信東受信

- ・  $PE_{ij}$  は  $W/N$ ワイヤ上で  $PE_{i,j+1}$  に送信する
- ・  $PE_{ij}$  は  $E/S$ ワイヤ上で  $PE_{i+1,j}$  から受信する

【0054】すべての場合に、競合は生ぜず、かつ適正な宛先点は  $PE$  の方形メッシュ接続構成のものと同じままである。

【0055】＜マトリックスの転置＞

【0056】マトリックス「 $P$ 」の転置の際に、マトリックスのマトリックスの行ベクトルが転置マトリックス「 $P^T$ 」の列ベクトルになる。マトリックス「 $P$ 」の任意の要素  $P_{ij}$  は転置マトリックス「 $P^T$ 」の要素  $P_{ji}$  になる。対角要素に変化はない。オラクルにおいて、マトリックス「 $P$ 」及びその転置を簡単に選択することができるが、これは要素  $P_{ij}$  と対応する要素  $P_{ji}$  の両方が2重要素に存在しているからである。「 $p$ 」レジスタの出力のセレクトは指定の操作において「 $P$ 」または「 $P^T$ 」の要素を使用することを可能とする。たとえば、「 $P$ 」マトリックスは図18に示されており、図19に示すように、オラクルにロードされる。

【0057】マトリックス  $P$  及び  $P^T$  の転置は図20に示されており、図21に示すように、オラクルにロードされる。

【0058】図22は畳み込みウィンドウを示す。図23ないし図26は方形メッシュにおけるイメージ処理畳み込みを示す。本明細書で検討するイメージ処理タスクの場合、 $3 \times 3$ の畳み込みウィンドウを備えた2-D畳み込み(図22)を想定する。メッシュ構造に用いられる技法は  $S. Y. Lee$  及び  $J. K. Aggarwal$  が提案したものである。処理要素の内部構造は図2に論理的に示されているが、これは  $Lee$  を改変したものであって、 $N, S, E, W$  の送信/受信ポートが詳細に示されている。

【0059】 $N \times N$ のイメージが  $N \times N$ のメッシュ  $P_{ij}$  レジスタにロード済みであり、 $R^i_{ij}$  レジスタがゼロにセットされているものと想定すると、畳み込みアルゴリズムを最初のウィンドウ要素  $W_{ij}$  から開始することができる。それ以降のステップは  $P_{ij}$  畳み込みパスにしたがうものであり、パス内の各ステップは図23ないし図26に対応するステップ番号を付けて示されている。(すべての  $PE$  がそのピクセル値で同じ演算を計算し、小計値を受け取ることに留意すべきである。)

(9)

特開平6-75986

16

【0060】1. 同報  $W_{ij}$  及び  $PE_{i,j+1}$  は  $R1 = 0 + W_{ij} P_{i,j+1}$  を計算し、 $R1$  北に転送する。

2. 同報  $W_{ij}$  及び  $PE_{i-1,j+1}$  は  $R2 = R1 + W_{ij} P_{i-1,j+1}$  を計算し、 $R2$  北に転送する。

3. 同報  $W_{ij}$  及び  $PE_{i-1,j}$  は  $R3 = R2 + W_{ij} P_{i-1,j}$  を計算し、 $R3$  東に転送する。

4. 同報  $W_{ij}$  及び  $PE_{i,j}$  は  $R4 = R3 + W_{ij} P_{i,j}$  を計算し、 $R3$  東に転送する。

5. 同報  $W_{ij}$  及び  $PE_{i,j+1}$  は  $R5 = R4 + W_{ij} P_{i,j+1}$  を計算し、 $R5$  南に転送する。

6. 同報  $W_{ij}$  及び  $PE_{i,j}$  は  $R6 = R5 + W_{ij} P_{i,j}$  を計算し、 $R6$  南に転送する。

7. 同報  $W_{ij}$  及び  $PE_{i,j+1}$  は  $R7 = R6 + W_{ij} P_{i,j+1}$  を計算し、 $R7$  西に転送する。

8. 同報  $W_{ij}$  及び  $PE_{i,j}$  は  $R8 = R7 + W_{ij} P_{i,j}$  を計算し、 $R8$  北に転送する。

9. 同報  $W_{ij}$  及び  $PE_{ij}$  は  $R9 = R8 + W_{ij} P_{ij}$  を計算し、停止する。

【0061】9つのステップの終了時に、各  $PE_{ij}$  は次のものを含むようになる。

$$\begin{aligned} \text{【数4】 } P_{ij} = & R9 = W_{ij} P_{i,j+1} + \\ & W_{ij} P_{i-1,j+1} + \\ & W_{ij} P_{i-1,j} + W_{ij} P_{i,j} + \\ & W_{ij} P_{i,j+1} + W_{ij} P_{i,j} + \\ & W_{ij} P_{i,j+1} + W_{ij} P_{ij} \end{aligned}$$

【0062】同様に、オラクルにおけるピクセル  $P_{ij}$  に対する典型的な畳み込み演算が、図27ないし図31に示されている。 $N \times N$ のイメージがオラクルの  $P_{ij}$  レジスタにロード済みであり、 $R^i_{ij}$  レジスタがゼロにセットされているものと想定すると、畳み込みアルゴリズムを最初のウィンドウ要素  $W_{ij}$  の同報から開始することができる。それ以降のステップは  $P_{ij}$  畳み込みパスにしたがうものであり、パス内の各ステップは図27ないし図31に対応するステップ番号を付けて示されている。

(すべての  $PE$  がそのピクセル値で同じ演算を計算し、小計値を受け取ることに留意すべきである。)

【0063】1. 同報  $W_{ij}$  及び  $PE_{i-1,j+1}$  は  $R1 = 0 + W_{ij} P_{i-1,j+1}$  を計算し、 $N/W$ ワイヤ上で  $R1$  に転送する。

2. 同報  $W_{ij}$  及び  $PE_{i,j}$  は  $R2 = R1 + W_{ij} P_{i,j}$  を計算し、 $N/W$ ワイヤ上で  $R2$  に転送する。

3. 同報  $W_{ij}$  及び  $PE_{i-1,j}$  は  $R3 = R2 + W_{ij} P_{i-1,j}$  を計算し、 $E/S$ ワイヤ上を  $R3$  に転送する。

4. 同報  $W_{ij}$  及び  $PE_{i,j}$  は  $R4 = R3 + W_{ij} P_{i,j}$  を計算し、 $E/S$ ワイヤ上を  $R4$  に転送する。

5. 同報  $W_{ij}$  及び  $PE_{i,j+1}$  は  $R5 = R4 + W_{ij} P_{i,j+1}$  を計算し、 $S/E$ ワイヤ上を  $R5$  に転送する。

6. 同報  $W_{ij}$  及び  $PE_{i,j}$  は  $R6 = R5 + W_{ij} P_{i,j}$  を計算し、 $S/E$ ワイヤ上を  $R6$  に転送する。

7. 同報  $W_{ij}$  及び  $PE_{i,j+1}$  は  $R7 = R6 + W_{ij} P_{i,j+1}$

50

(10)

特開平6-75986

17

18

を計算し、W/Nワイヤ上をR7に転送する。

8. 同報W<sub>8</sub>及びP E<sub>10</sub>はR8=R7+W<sub>8</sub>P<sub>10</sub>を計算し、N/Wワイヤ上をR8に転送する。

9. 同報W<sub>8</sub>及びP E<sub>10</sub>をR9=R8+W<sub>8</sub>P<sub>10</sub>を計算し、停止する。

【0064】9つのステップの終了時に、各P E<sub>10</sub>は次のものを含むようになる。

【数5】P<sub>10</sub>=R9=W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>

W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>

W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>

W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>

W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>+W<sub>8</sub>P<sub>10</sub>

【0065】ピクセルP45の場合、これは次のように\*

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial}{\partial y} \frac{\partial u}{\partial x} + C \frac{\partial^2 u}{\partial y^2} + D \frac{\partial u}{\partial x} + E \frac{\partial u}{\partial y} + F u = G \quad (1)$$

【0069】ここで、A、B、C、D、E、F及びGはx及びyの関数であり、領域Rで連続し、Sが境界となる。関数u(x, y)はR及びSで連続していなければならない。

【0070】有限差分法において、メッシュは領域Rに\*

$$\frac{\partial u}{\partial x} \approx [u(x+h_x, y) - u(x-h_x, y)]/(2h_x) \quad (2a)$$

【0071】 ★ ★ 【数9】

$$\frac{\partial u}{\partial y} \approx [u(x, y+h_y) - u(x, y-h_y)]/(2h_y) \quad (2b)$$

【0072】 ☆ ☆ 【数10】

$$\frac{\partial^2 u}{\partial x^2} \approx [u(x+h_x, y) + u(x-h_x, y) - 2u(x, y)]/h_x^2 \quad (2c)$$

【0073】 ◆ ◆ 【数11】

$$\frac{\partial^2 u}{\partial y^2} \approx [u(x, y+h_y) + u(x, y-h_y) - 2u(x, y)]/h_y^2 \quad (2d)$$

【0074】ただし、h<sub>x</sub>及びh<sub>y</sub>はそれぞれx及びy軸のメッシュ間隔である(図33)。通常、水平及び垂直両方高のメッシュ間隔は同じである(式3)。

【数12】h=h<sub>x</sub>=h<sub>y</sub> (3)

【0075】=0として、式(2a)ないし(2d)を式(1)に代入し、-h<sup>2</sup>を乗じると、次の式が得られる。

【数13】a<sub>1</sub>u(x, y)-a<sub>1</sub>u(x+h, y)-a<sub>1</sub>u(x, y+h)-a<sub>1</sub>u(x-h, y)-a<sub>1</sub>u(x, y-h)=t(x, y)

【0076】ただし、

【数14】

$$a_1 = A(x, y) + \frac{h}{2} D(x, y)$$

\*なる。

【数6】G(P45)=W11P36+W12P35+W13P34+W23P44+W33P54+W32P55+W31P56+W21P46+W22P45

【0066】<有限差分法の例>

【0067】微分方程式を解くための有限差分法は物理学、機械工学、及び電気工学などの多数の科学技術分野で広く用いられている。このような方法において、微分方程式の導関数は、切り捨てられたテイラー級数から得られる差分商によって近似される。

【0068】有限差分法において、2次偏微分方程式、式(1)を考える。

【数7】

\*重ねられ(図32に示すように)、微分方程式(1)は差分方程式に置換される。偏導関数は中心差分商方程式2a-2d(図33)に置換される。

20 【数8】

【0077】

【数15】

$$a_2 = C(x, y) + \frac{h}{2} E(x, y)$$

【0078】

【数16】

$$a_3 = A(x, y) + \frac{h}{2} D(x, y)$$

【0079】

【数17】

$$a_4 = C(x, y) + \frac{h}{2} E(x, y)$$



(11)

特開平6-75986

19

20

【0080】

【数18】  $a_0 = a_1 + a_2 + a_3 + a_4 - h^2 F(x, y)$ 

【0081】

【数19】  $t(x, y) = -h^2 G(x, y)$ 

【0082】ラプラスの方程式を考えると、次のようになる。

【数20】

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

【0083】ただし、 $A=C=1$ 及び $B=D=E=F=G=0$ である。

【0084】領域Rに境界条件Sを与えると、一意の解が得られる。ラプラスの方程式を小さな差分として表すことができる。代入により、式(7a)が得られる。

【数21】  $4u(x, y) - u(x+h, y) - u(x, y+h) - u(x-h, y) - u(x, y-h) = 0$ 【0085】それ故、 $u(x, y)$ の値を式(7b)を使用した反復法によって計算できる。【数22】  $u(x, y) = [u(x+h, y) + u(x, y+h) + u(x-h, y) + u(x, y-h)] / 4$ 

【0086】ラプラスの方程式及びポアソンの方程式は、振動膜の解析などの多くの理論及び応用問題に広く使用されている。

【0087】図5ないし図14の説明で提示したオラクルの表記を使用して、式(7a)を式8にマップすることによって、式(7b)をオラクルについて計算することができる。

【数23】  $P(i, j) = [P(i+1, j) + P(i, j+1) + P(i-1, j) + P(i, j-1)] / 4$ 

【0088】この表記は連続性に合わせて連続するものであるが、PEの内部構造(図5)は有限差分法アルゴリズムに合わせて修正される。修正されたPEは図34に示されている。対角PEも同様に、図10に示す形態から2重PE構造に修正される。対角PEは1つの実行ユニットと、単一PEに適したレジスタを含んでいるが、その構造は図34から推論されたものである。同一の送受入力ポート及び制御が新しいPE内部構造に利用されているが、計算機能は修正されている。アキュムレータ/シフタを備えた加算器を利用して、式8に必要な4回の演算によって加算及び除算を行う。Wレジスタは比較論理が使用する最大許容誤差値を格納する。新しい $P(i, j)$ 値を計算した後、以前の $P(i, j)$ 値と比較し、いずれかのPEで差が最大許容誤差よりも大きい場合には、計算を継続する必要がある。すべてのPEの誤差値が最大許容誤差よりも少なくなければならないのであるから、制御システムはPEの

比較演算の状態を知っていなければならない。これは論理値を各PEからメッシュ構造を介して、制御システムへ大域解析のために送ることによって得られる。図35ないし図38は北、南、東及び西の伝送モードのためのスイッチ及び入出力ポートの構成を示す。

【0089】PEレジスタにロードされる初期値は問題によって異なっている。アキュムレータ $R^+$ はゼロに初期設定され、最大許容誤差値W。レジスタにロードされる。初期設定後、以下のステップが行われるが、これらのステップについては図39ないし図41のステップ1ないしステップ5を参照されたい。

【0090】1. ステップ1。PEの値を北へ送信し、受信した値を $R^+$ の値に加える。

2. ステップ2。PEの値を東へ送信し、受信した値を $R^+$ の値に加える。

3. ステップ3。PEの値を南へ送信し、受信した値を $R^+$ の値に加える。

4. ステップ4。PEの値を西へ送信し、受信した値を $R^+$ の値に加える。

5. ステップ5。 $R^+$ に蓄積した値を2ポジション右へシフトし、4で除算し、シフトした蓄積値 $R^+$ を元の値PEと比較して、2つの値が最大許容誤差内にあるかどうかを検証する。比較の結果をアレイの縁部へ送り、収束の全体的な判断を行う。

【0091】全体的な収束にいたっていない場合には、収束が全体的に達成されるまで、上記のプロセスを継続する。

【0092】

【発明の効果】本発明により、並列アレイ・コンピュータ・システムすなわち大規模並列アレイ・プロセッサに用いることのできるコンピュータ・システムの新規なPE及び関連した編成を提供できる。さらに、微分方程式を解くのに有限差分法を使用することを含む、マルチメディア用及び汎用大規模並列コンピュータ・システムを提供できる。

【図面の簡単な説明】

【図1】畳み込みイメージ処理用の $8 \times 8$ のメッシュを示す図面である。

【図2】メッシュPE内部構造の図面である。

【図3】本発明の好ましい実施例による $8 \times 8$ のメッシュを示す図面である。

【図4】本発明の好ましい実施例による $8 \times 8$ のメッシュを示す図面である。

【図5】転置及び畳み込みをサポートする対称PE「セル」構造の図面である。

【図6】転置及び畳み込みをサポートする対称PE「セル」構造の図面である。

【図7】転置及び畳み込みをサポートする対称PE「セル」構造の図面である。

【図8】転置及び畳み込みをサポートする対称PE「セ



(12)

特開平6-75986

21

ル」構造の図面である。

【図9】転置及び畳み込みをサポートする対称PE「セル」構造の図面である。

【図10】転置及び畳み込みをサポートする対称PE「セル」構造の図面である。

【図11】転置及び畳み込みをサポートする対称PE「セル」構造の図面である。

【図12】転置及び畳み込みをサポートする対称PE「セル」構造の図面である。

【図13】転置及び畳み込みをサポートする対称PE「セル」構造の図面である。

【図14】転置及び畳み込みをサポートする対称PE「セル」構造の図面である。

【図15】本発明の好ましいPEの記号体系を付した8×8のメッシュの図面である。

【図16】本発明の好ましいPEの記号体系を付した8×8のメッシュの図面である。

【図17】本発明の好ましいPEの記号体系を付した8×8のメッシュの図面である。

【図18】方形メッシュにおける8×8のマトリックスPの図面である。

【図19】本発明の好ましい実施例における8×8のマトリックスPの図面である。

【図20】方形メッシュにおける8×8のマトリックスと転置したマトリックスPの図面である。

【図21】本発明の好ましい実施例における8×8のマトリックスと転置したマトリックスPの図面である。

【図22】畳み込みウィンドウの図面である。

【図23】畳み込みイメージ処理のための一般化されたメッシュの図面である。

【図24】畳み込みイメージ処理のための一般化されたメッシュの図面である。

【図25】畳み込みイメージ処理のための一般化された\*

22

\*メッシュの図面である。

【図26】畳み込みイメージ処理のための一般化されたメッシュの図面である。

【図27】ピクセルP45の畳み込みのステップ1及び3を示す図面である。

【図28】ピクセルP45の畳み込みのステップ2及び4を示す図面である。

【図29】ピクセルP45の畳み込みのステップ5及び7を示す図面である。

【図30】ピクセルP45の畳み込みのステップ6及び8を示す図面である。

【図31】ピクセルP45の畳み込みのステップ9を示す図面である。

【図32】領域Rに重ねられたメッシュを示す図面である。

【図33】領域Rに重ねられたメッシュを示す図面である。

【図34】有限差分法をサポートする本発明の対称PE「セル」を示す図面である。

【図35】有限差分法をサポートする本発明の対称PE「セル」を示す図面である。

【図36】有限差分法をサポートする本発明の対称PE「セル」を示す図面である。

【図37】有限差分法をサポートする本発明の対称PE「セル」を示す図面である。

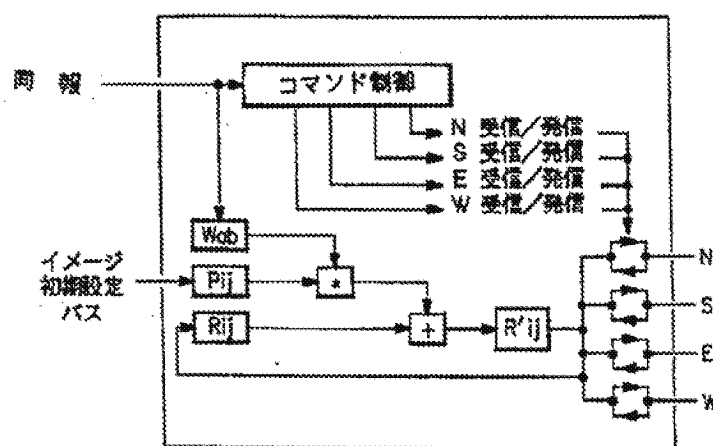
【図38】有限差分法をサポートする本発明の対称PE「セル」を示す図面である。

【図39】本発明の有限差分法のステップ1及び3を示す図面である。

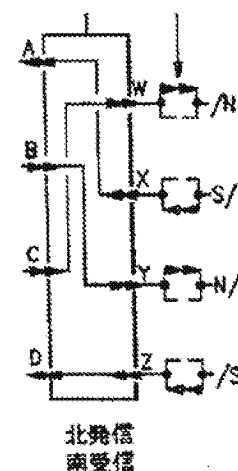
【図40】本発明の有限差分法のステップ2及び4を示す図面である。

【図41】本発明の有限差分法のステップ5を示す図面である。

【図2】



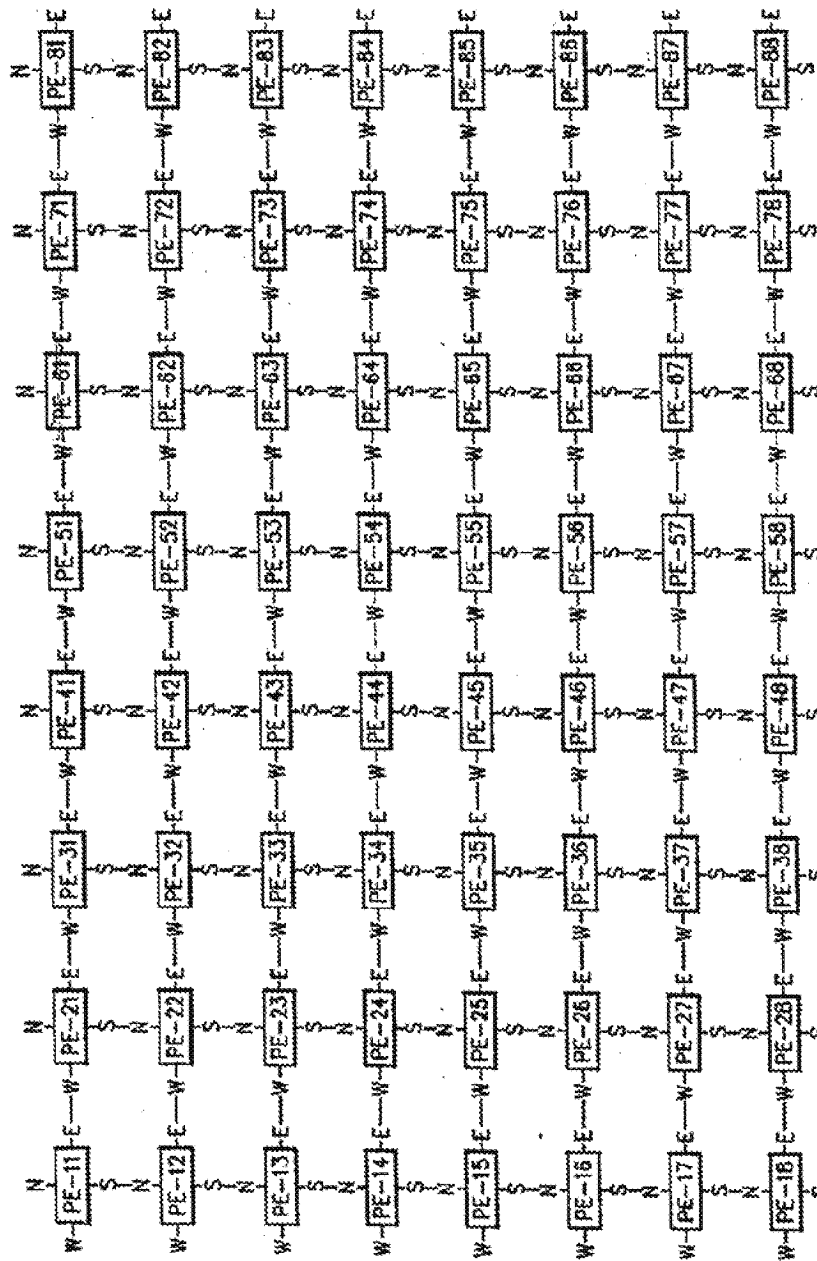
【図6】



(13)

特開平6-75986

【図1】

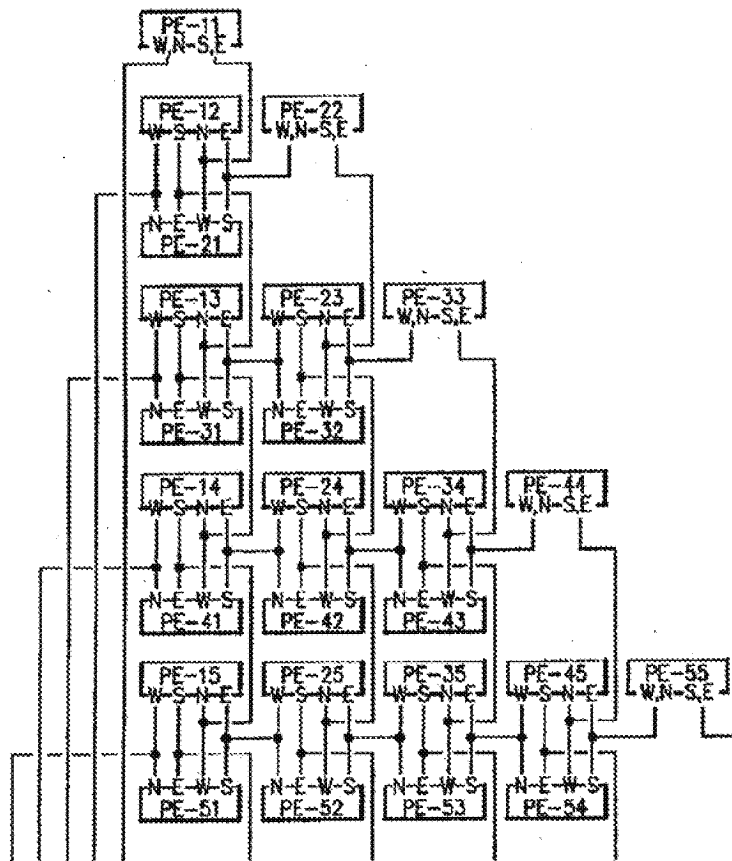


2Nの折り返し模様

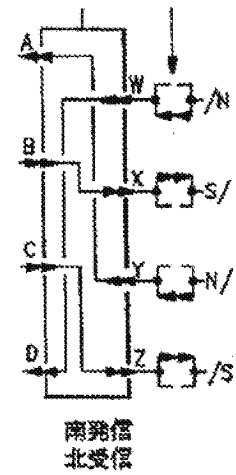
(14)

特開平6-75986

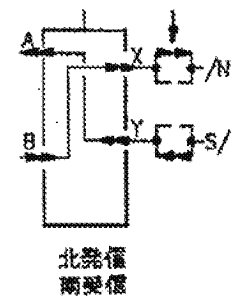
【図3】



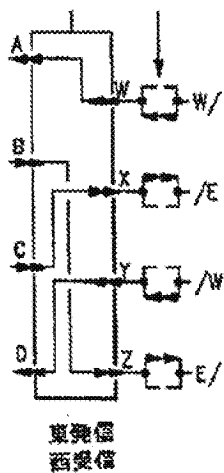
【図7】



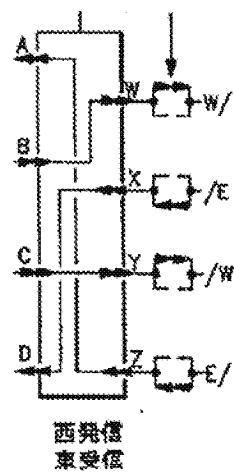
【図11】



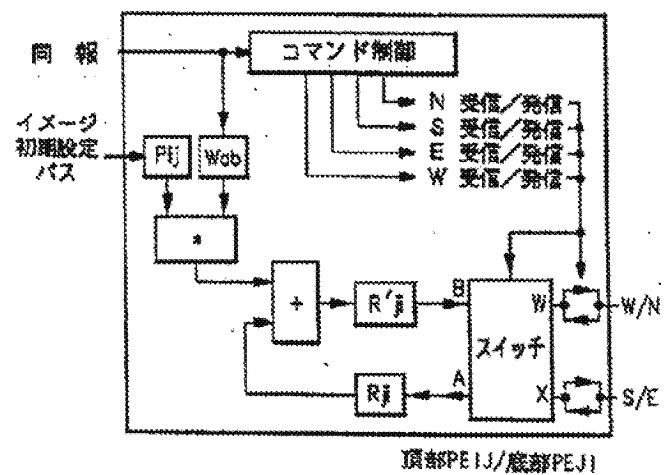
【図8】



【図9】



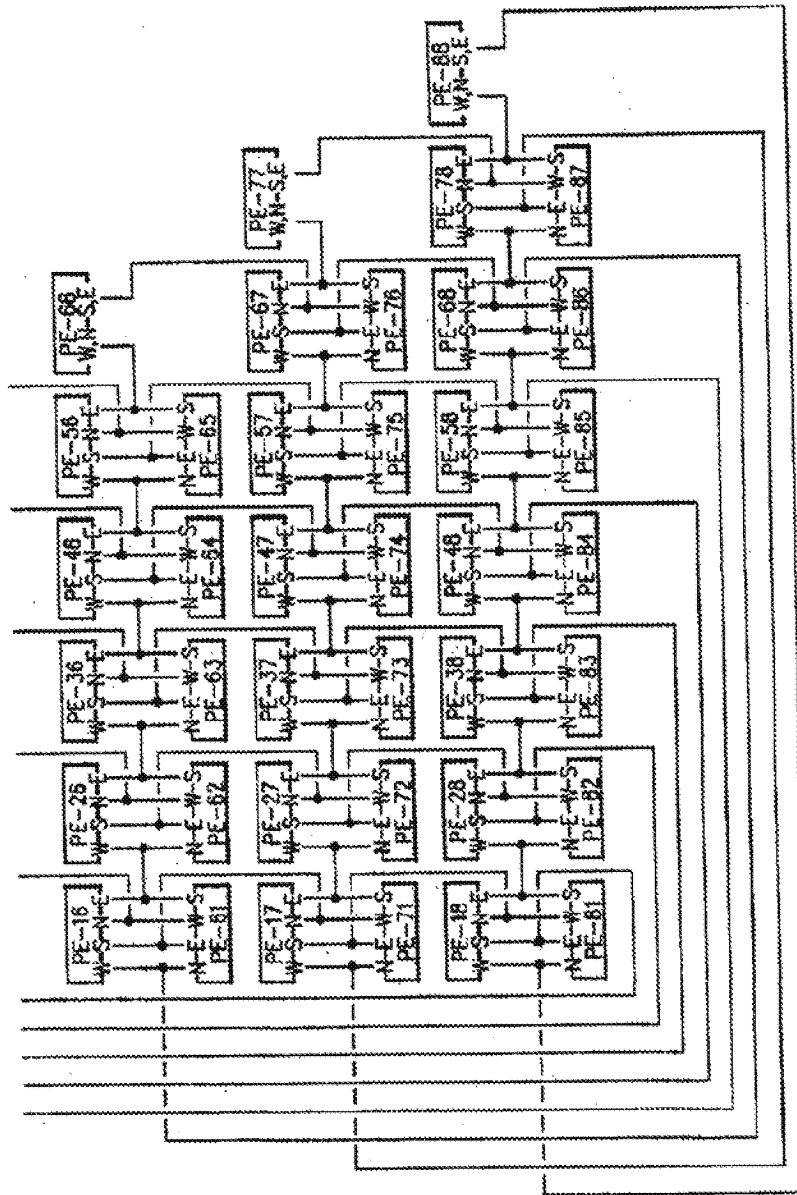
【図10】



(15)

特開平6-75986

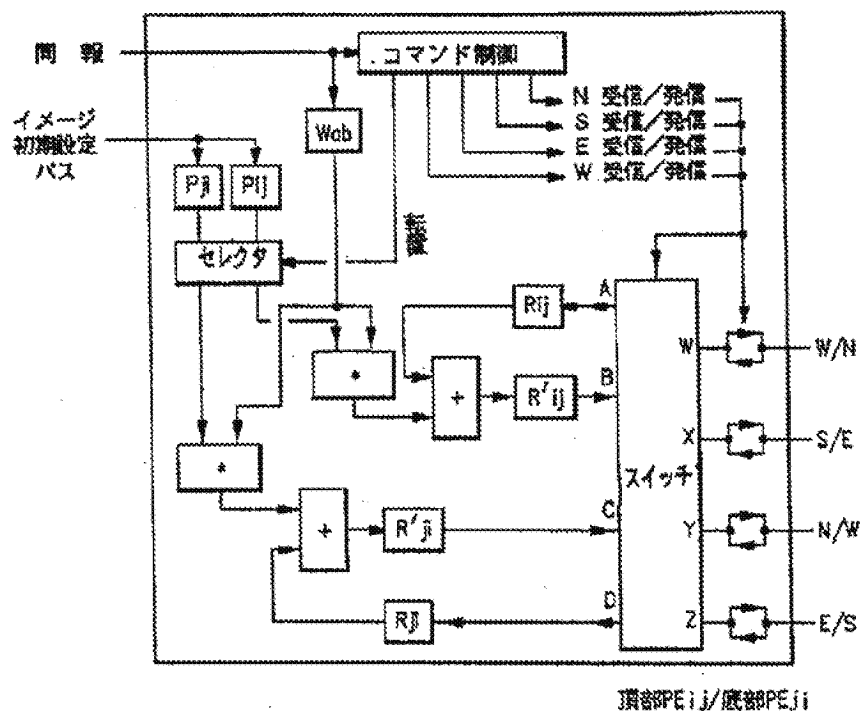
【図4】



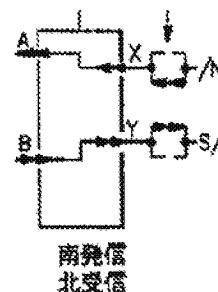
(16)

特開平 6-75986

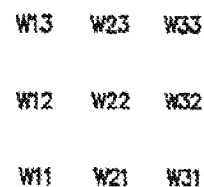
[EX 5]



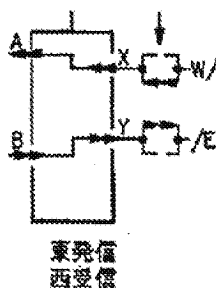
【例 12】



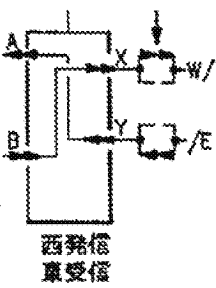
【例 22】



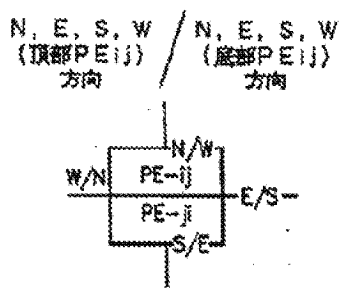
【例 13】



【例 4】



【例 15】



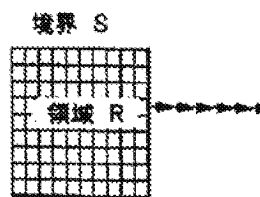
【圖 8】

[illegible]

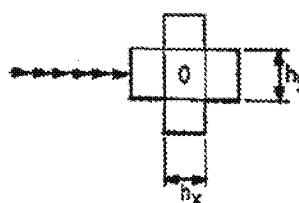
【例20】

1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8

321



【图 3-3】

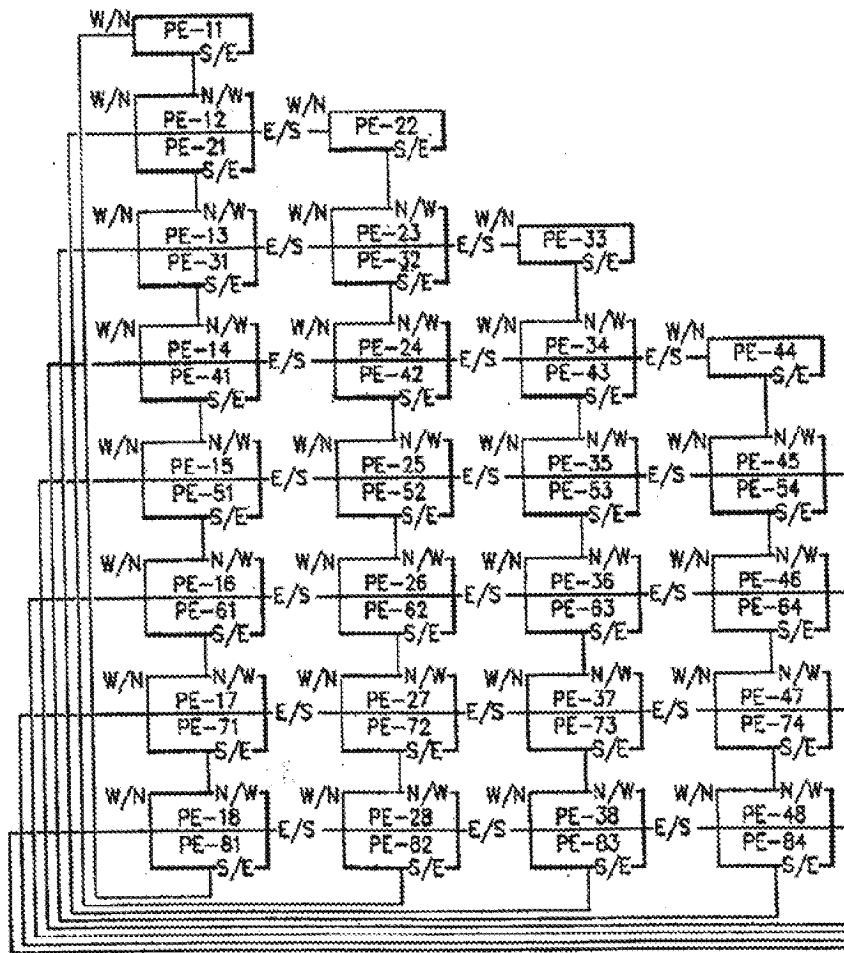




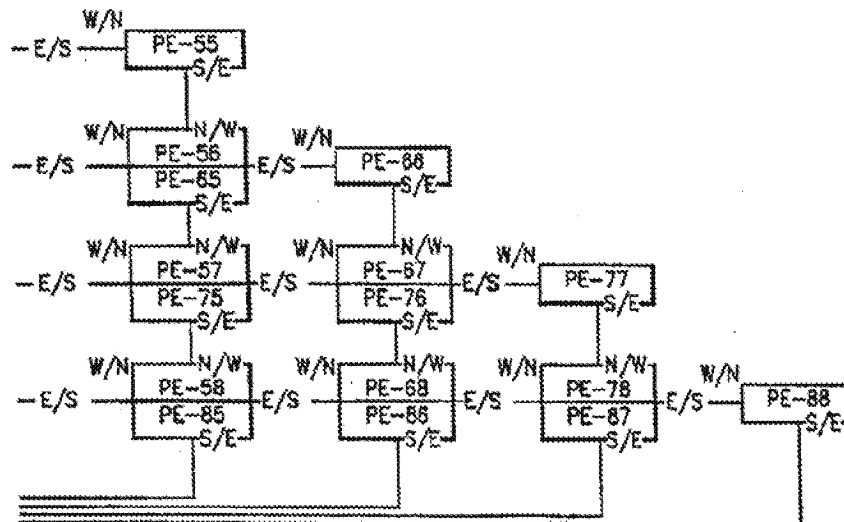
(17)

特開平6-75986

【図16】



【図17】



【図19】

1
1 2
2
1 2 3
3 3
1 2 3 4
4 4 4
1 2 3 4 5
5 5 5 5
1 2 3 4 5 6
6 6 6 6 6
1 2 3 4 5 6 7
7 7 7 7 7 7
1 2 3 4 5 6 7 8
8 8 8 8 8 8 8

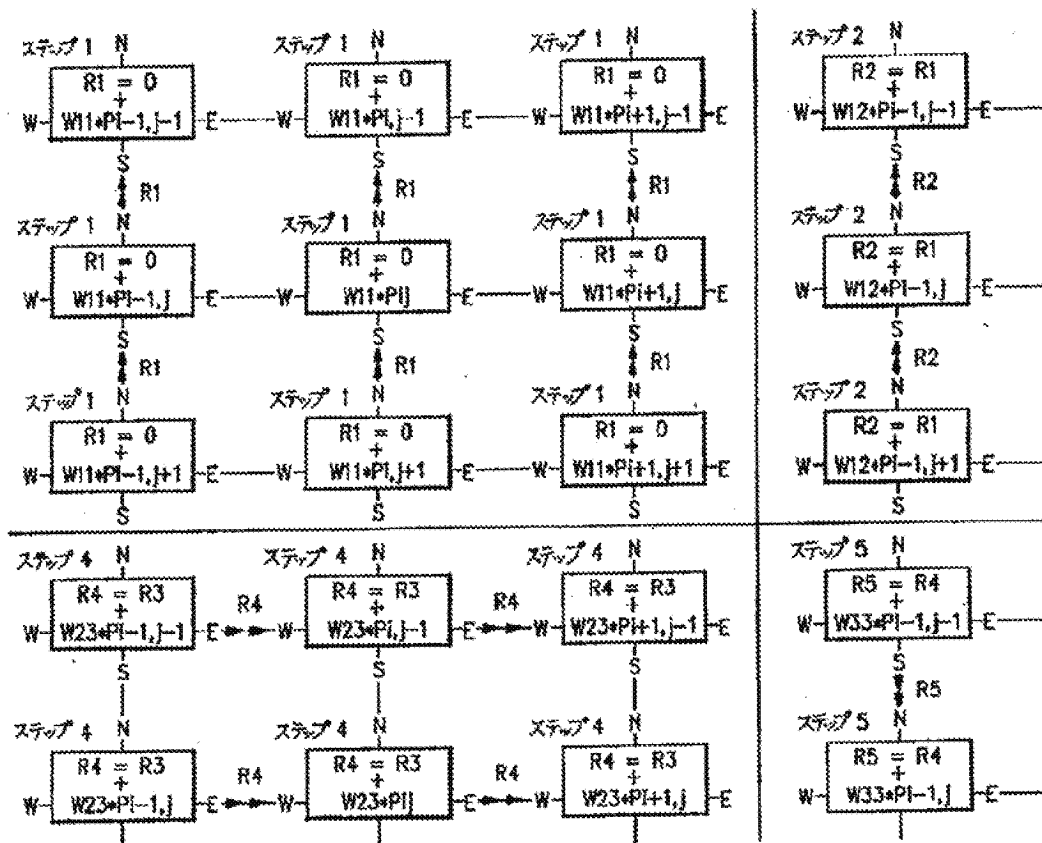
【図21】

1
2 2
1
3 3 3
1 2
4 4 4 4
1 2 3
5 5 5 5 5
1 2 3 4
6 6 6 6 6 6
1 2 3 4 5
7 7 7 7 7 7 7
1 2 3 4 5 6
8 8 8 8 8 8 8 8
1 2 3 4 5 6 7

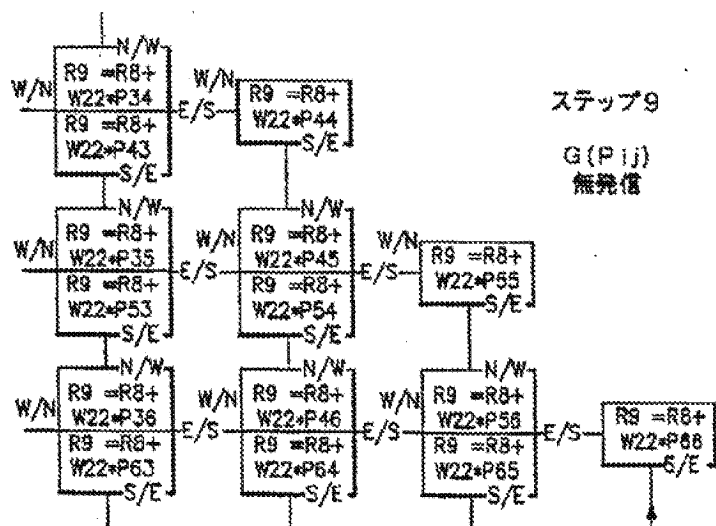
(18)

特開平6-75986

【図23】

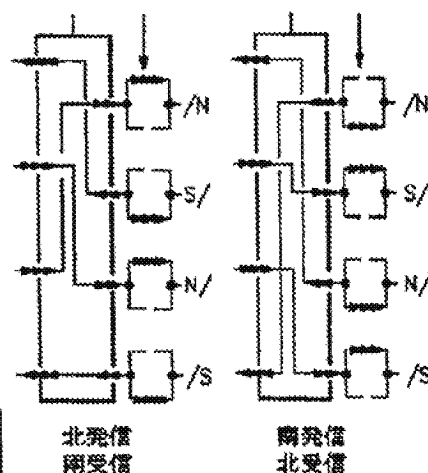


【図31】



【図35】

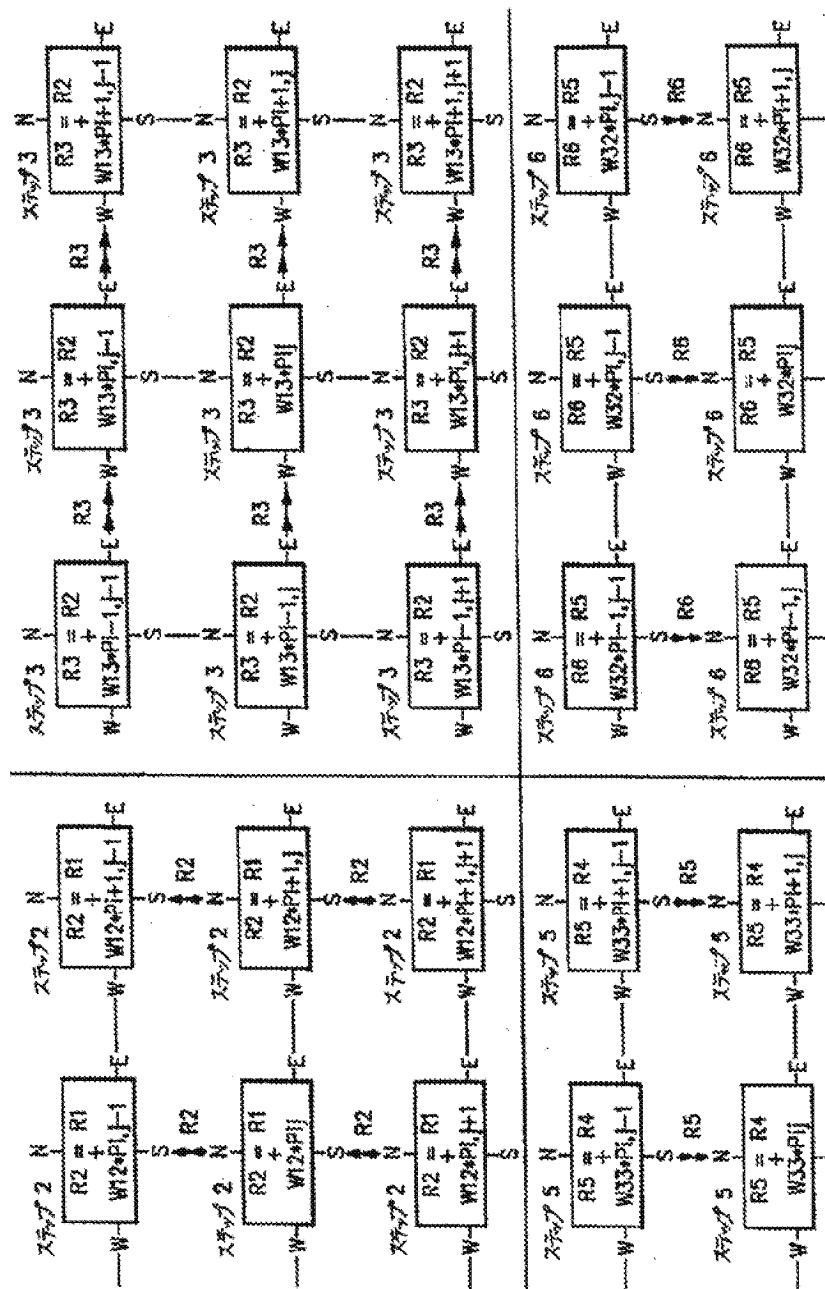
【図36】



(19)

表 6-75986

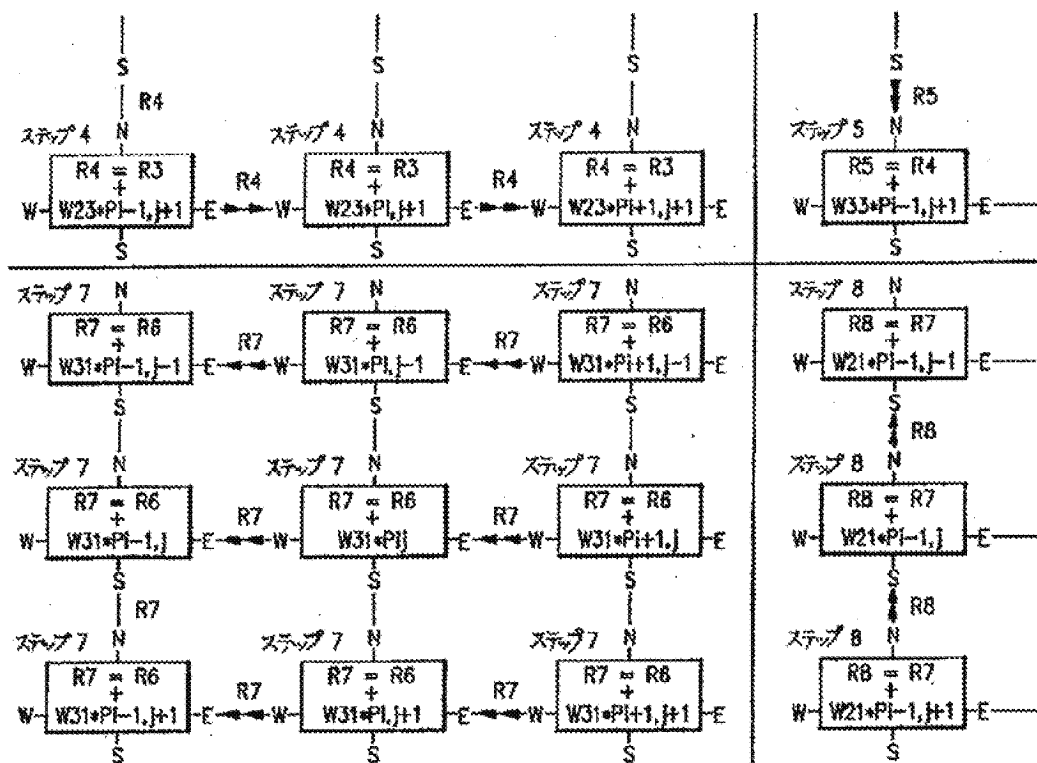
【図 24】



(20)

特開平6-75986

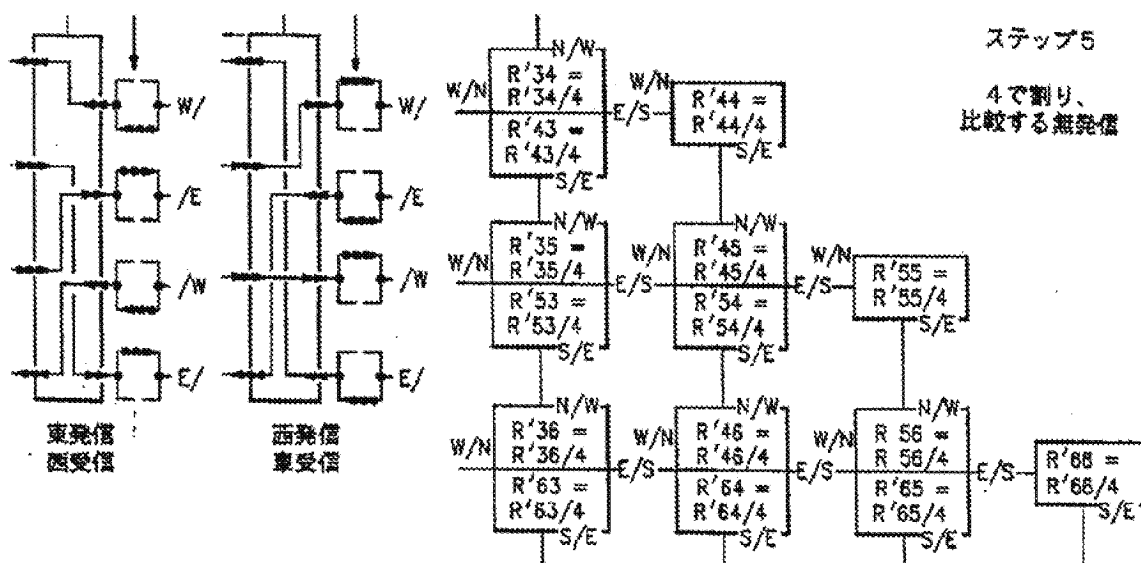
【図25】



【図37】

【図38】

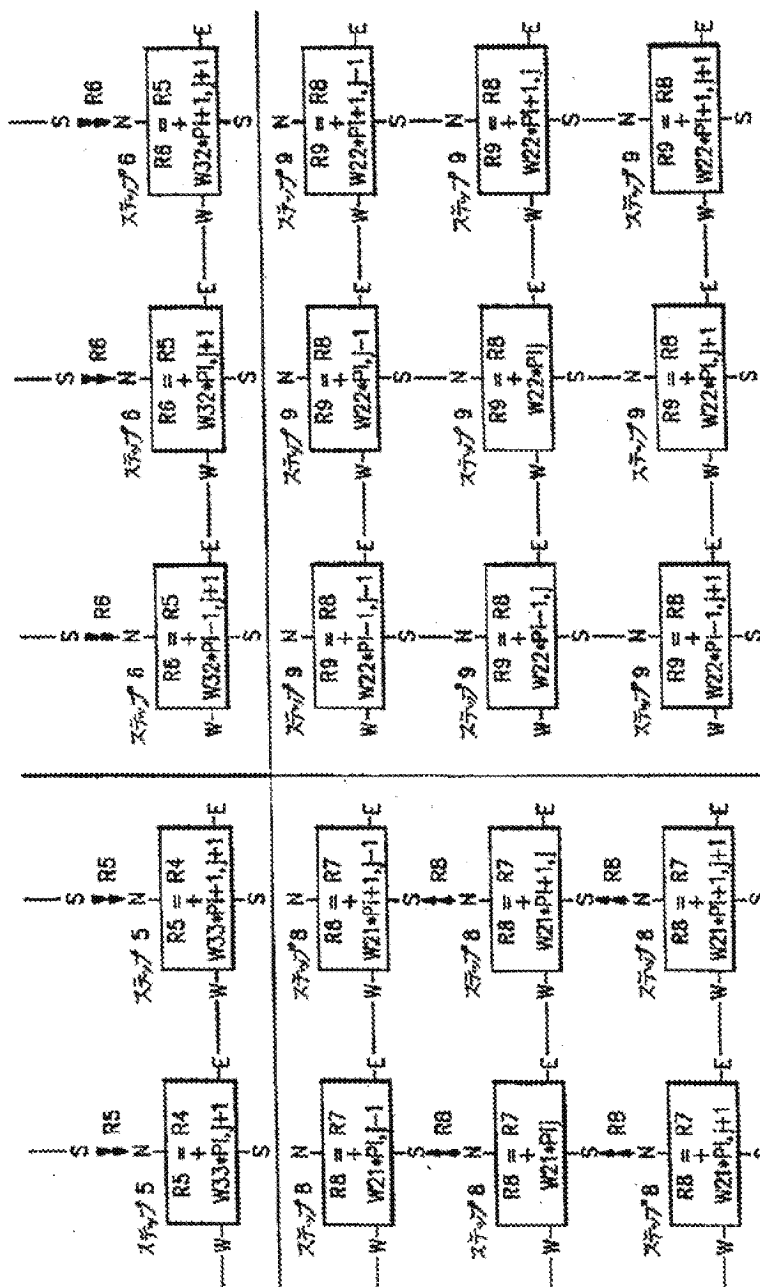
【図41】



(21)

特開平6-75986

【図26】

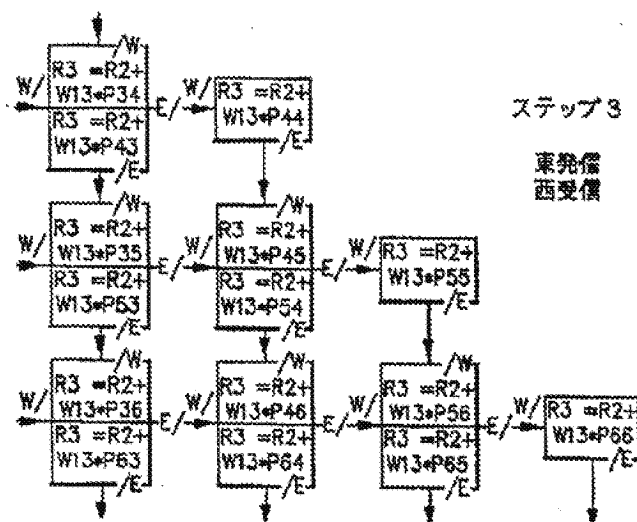
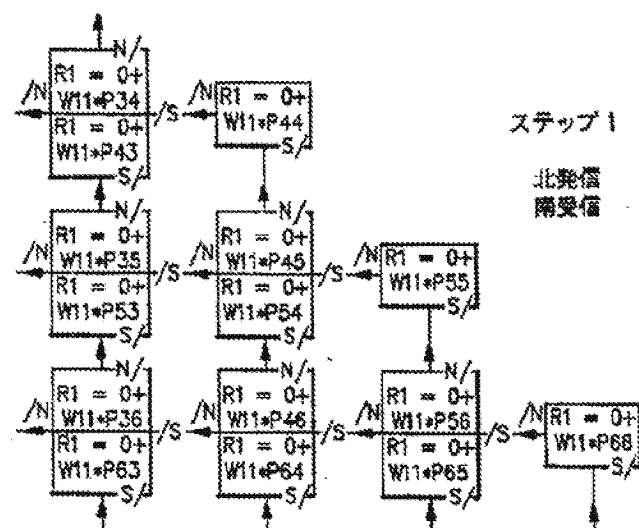




(22)

特開平6-75986

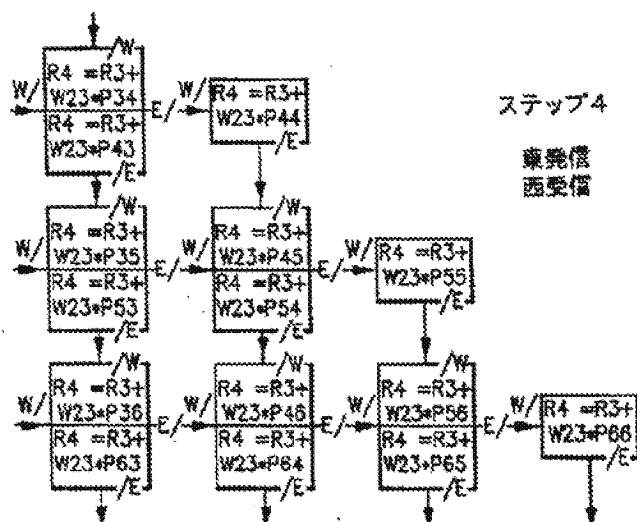
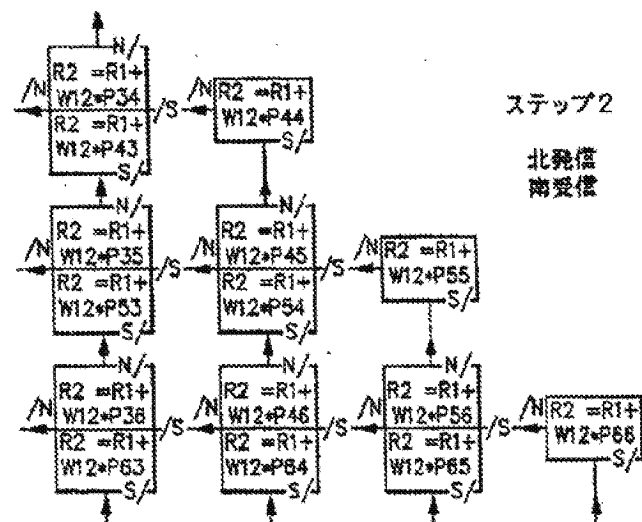
【図27】



(23)

特開平6-75986

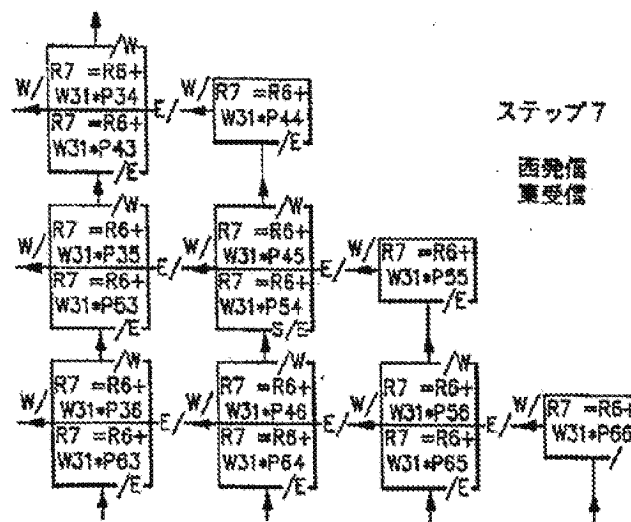
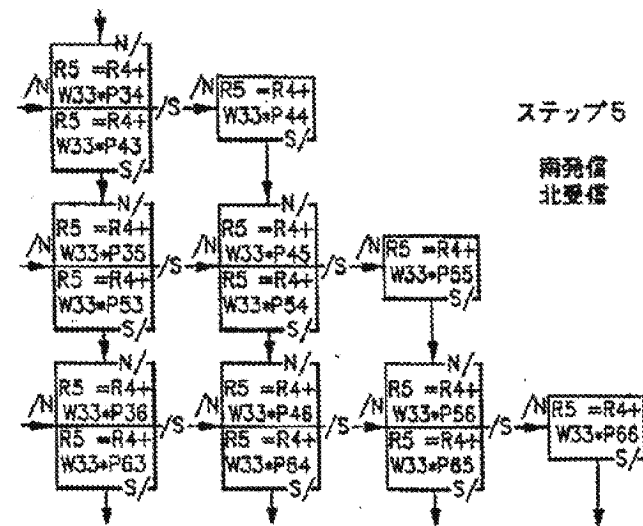
【図28】



(24)

特開平6-75986

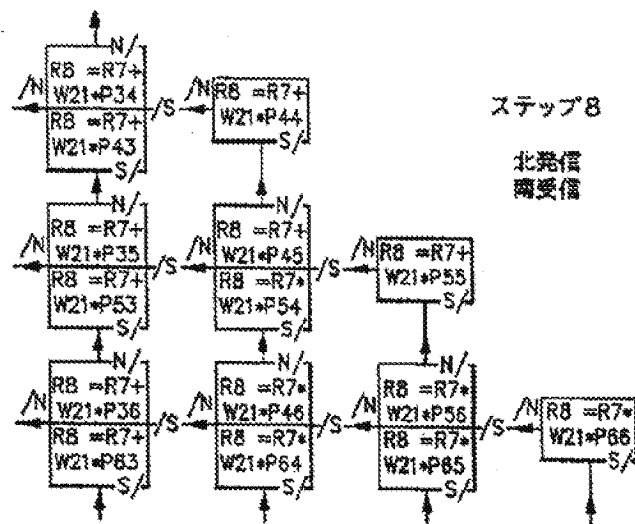
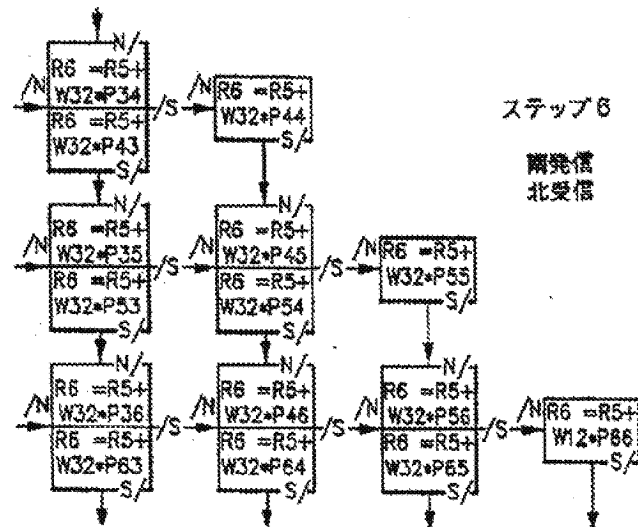
【図29】



(25)

特開平6-75986

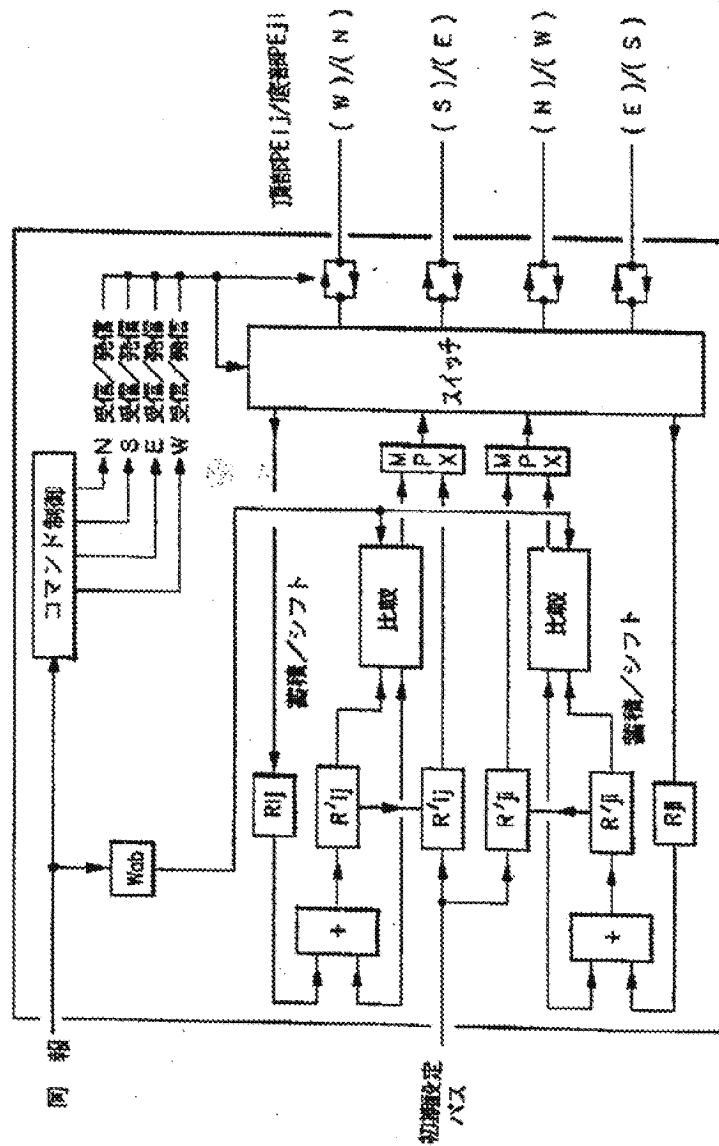
【図30】



(26)

特開平6-75986

【図34】

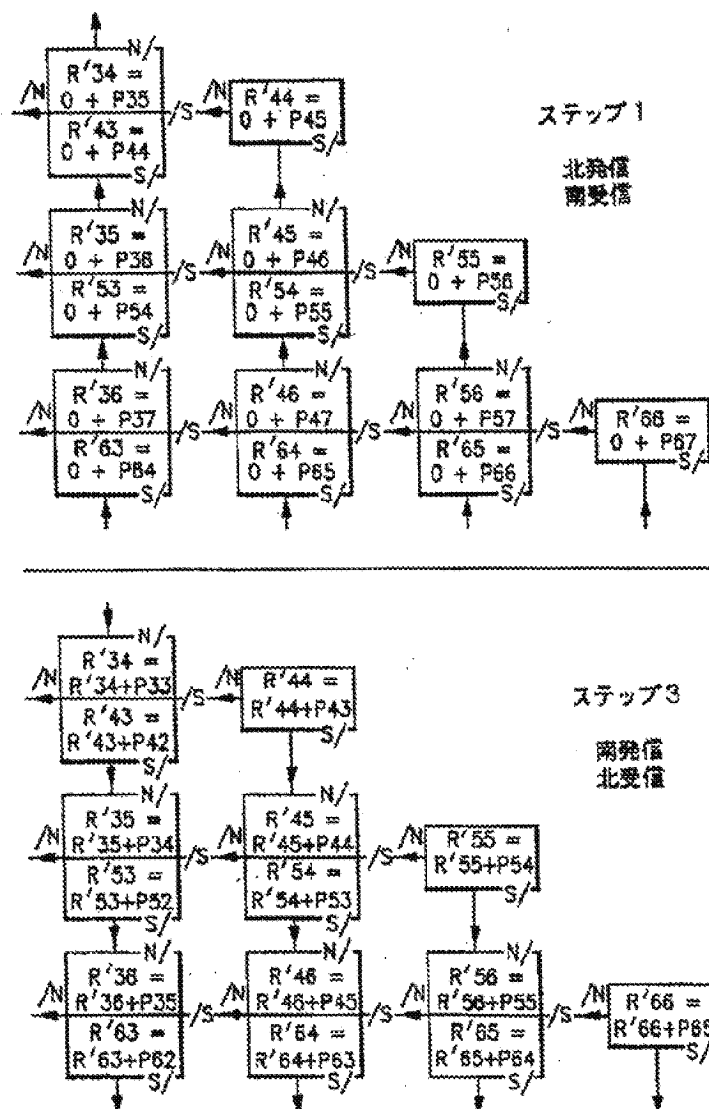




(27)

特開平6-75986

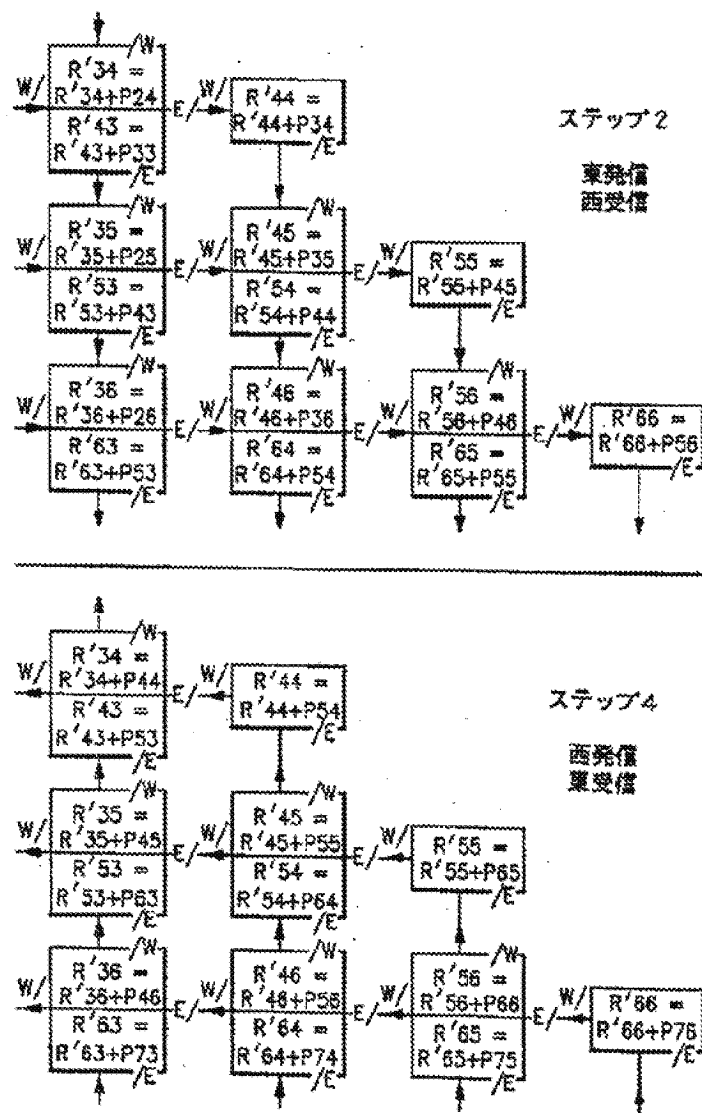
【図39】



(28)

特開平6-75986

【図40】



フロントページの続き

(72)発明者 スタマティス・ヴァッシリアディス  
13850ニューヨーク州ヴェスタル、ヴェス  
タル・ロード 717

(72)発明者 ホセ・グアダループ・デルガードーフリ  
アス  
13760ニューヨーク州エンドウエル、レー  
シー・ドライブ 612



US006405185B1

(12) **United States Patent**  
Pechanek et al.

(10) **Patent No.:** **US 6,405,185 B1**  
(45) **Date of Patent:** **Jun. 11, 2002**

(54) **MASSIVELY PARALLEL ARRAY  
PROCESSOR**

5,509,106 A \* 3/1996 Pechanek et al. .... 706/41

#### FOREIGN PATENT DOCUMENTS

EP 456201 A 11/1991  
GB 2219106 11/1989

#### OTHER PUBLICATIONS

IEEE, 1988, "Mapping Signal Processing Algorithm To Fixed Architectures", V2. 10, pp. 2037-2040, by Stewart.

(List continued on next page.)

Primary Examiner—George B. Davis

(74) Attorney, Agent, or Firm—John E. Hoel, Esq.; Morgan & Finnegan, LLP; Scott W. Reid, Esq.

(75) Inventors: Gerald George Pechanek, Endwell;  
Stamatis Vassiliadis, Vestal; Jose  
Guadalupe Delgado-Frias, Endwell, all  
of NY (US)

(73) Assignee: International Business Machines  
Corporation, Armonk, NY (US)

(\*) Notice: Subject to any disclaimer, the term of this  
patent is extended or adjusted under 35  
U.S.C. 154(b) by 0 days.

(21) Appl. No.: 09/551,144

(22) Filed: Mar. 23, 1995

#### Related U.S. Application Data

(63) Continuation of application No. 07/864,112, filed on Apr. 6,  
1992, now abandoned.

(51) Int. Cl.<sup>7</sup> ..... G06F 15/18; G06F 15/00

(52) U.S. Cl. .... 706/41; 706/14; 706/42;  
712/19

(58) Field of Search ..... 706/41, 14, 42;  
712/19

#### (56) References Cited

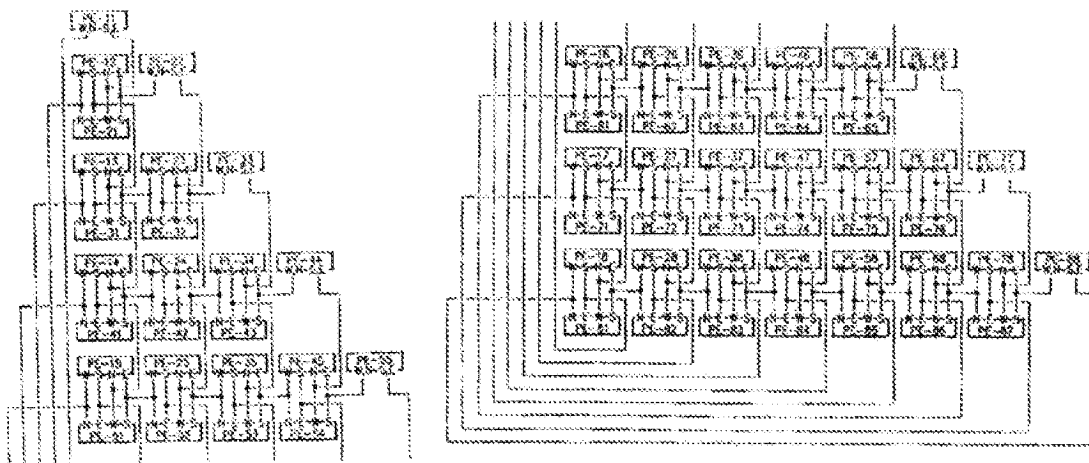
##### U.S. PATENT DOCUMENTS

4,270,170 A 5/1981 Reddaway ..... 712/11  
4,972,361 A 11/1990 Rader ..... 708/522  
5,018,065 A \* 5/1991 McWhirter et al. .... 712/19  
5,138,695 A \* 8/1992 Meaus et al. .... 706/41  
5,148,515 A \* 9/1992 Vassiliadis et al. .... 706/41  
5,274,632 A \* 12/1993 Khan ..... 706/424  
5,297,289 A \* 3/1994 Muntzer ..... 712/19

#### (57) ABSTRACT

Image processing for multimedia workstations is a computationally intensive task requiring special purpose hardware to meet the high speed requirements associated with the task. One type of specialized hardware that meets the computation high speed requirements is the mesh connected computer. Such a computer becomes a massively parallel machine when an array of computers interconnected by a network are replicated in a machine. The nearest neighbor mesh computer consists of an  $N \times N$  square array of Processor Elements (PEs) where each PE is connected to the North, South, East and West PEs only. The diagonal folded mesh array processor, which is called Oracle, allows the matrix transformation operation to be accomplished in one cycle by simple interchange of the data elements in the dual symmetric processor elements. The use of Oracle for a parallel 2-D convolution mechanism for image processing and multimedia applications and for a finite difference method of solving differential equations is presented, concentrating on the computational aspects of the algorithm.

4 Claims, 22 Drawing Sheets



US 6,405,185 B1

Page 2

OTHER PUBLICATIONS

Ward et al, "The application and development of wavefront array processors for advanced front-end signal processing systems"; Papers presented at the first international workshop on systolic arrays, oxford, p. 295-301, Jul. 2-4, 1986.\*  
Cappello, Peter "A Processor-Time-Minimal Systolic Array for Cubical Mesh Algorithms", IEEE Transactions on Parallel and Distributed Systems, Jan. 1992.\*  
IBM Technical Disclosure Bulletin, vol. 34, No. 10A, 3/92, pp.100-106, "Many-Snap".  
Computer Architecture News, vol. 20, No.2, 05/92, NY US; C. Glass & L. Ni; "The Turn Model for Adaptive Routing".

Signal Processing IV; Theories & Applications Proceedings of Eusipco-88 vol. 1, 9/88, Grenoble France; pp. 347-350, S. Kung & R. Stewart "A Triangular Array Processor for Kalman Filtering".

IEEE International Conf. on Computer Design: VLSI in Computers, 10/86 pp. 266-274, D. Koppelman et al; "The implementation of a triangular permutation network using wafer scale integration".

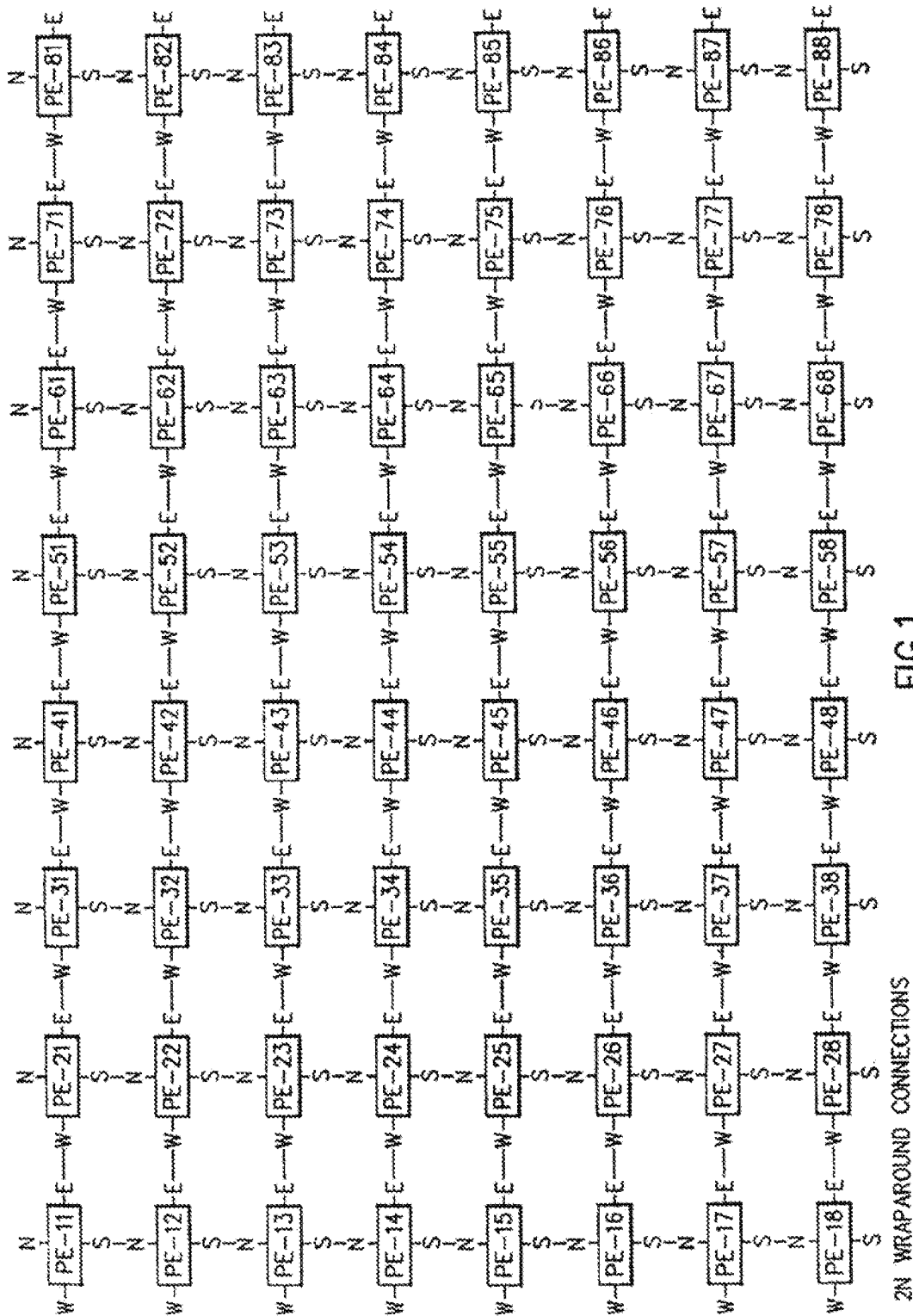
\* cited by examiner

U.S. Patent

Jun. 11, 2002

Sheet 1 of 22

US 6,405,185 B1





U.S. Patent

Jun. 11, 2002

Sheet 2 of 22

US 6,405,185 B1

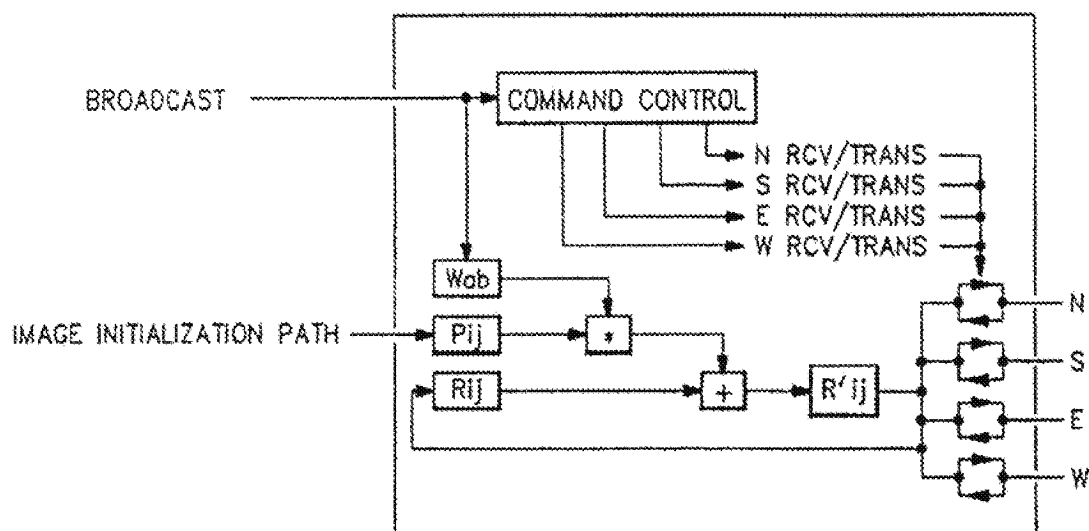


FIG. 2

1 2 3 4 5 6 7 8  
 1 2 3 4 5 6 7 8  
 1 2 3 4 5 6 7 8  
 1 2 3 4 5 6 7 8  
 1 2 3 4 5 6 7 8  
 1 2 3 4 5 6 7 8  
 1 2 3 4 5 6 7 8  
 1 2 3 4 5 6 7 8

FIG. 6A

1  
 1 2  
 2  
 1 2 3  
 3 3  
 1 2 3 4  
 4 4 4  
 1 2 3 4 5  
 5 5 5 5  
 1 2 3 4 5 6  
 6 6 6 6 6  
 1 2 3 4 5 6 7  
 7 7 7 7 7 7  
 1 2 3 4 5 6 7 8  
 8 8 8 8 8 8 8

FIG. 6B

FIG. 3A

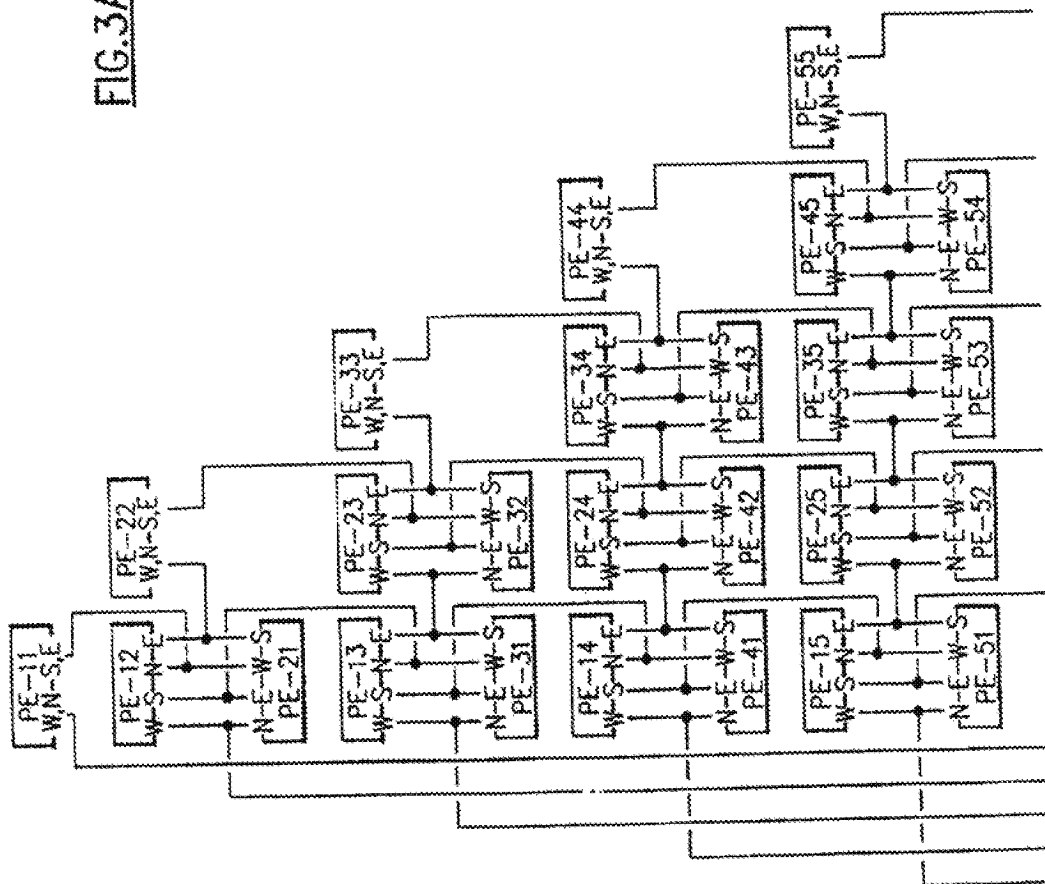


FIG. 3A

FIG. 58

FIG. 3

U.S. Patent

Jun. 11, 2002

Sheet 4 of 22

US 6,405,185 B1

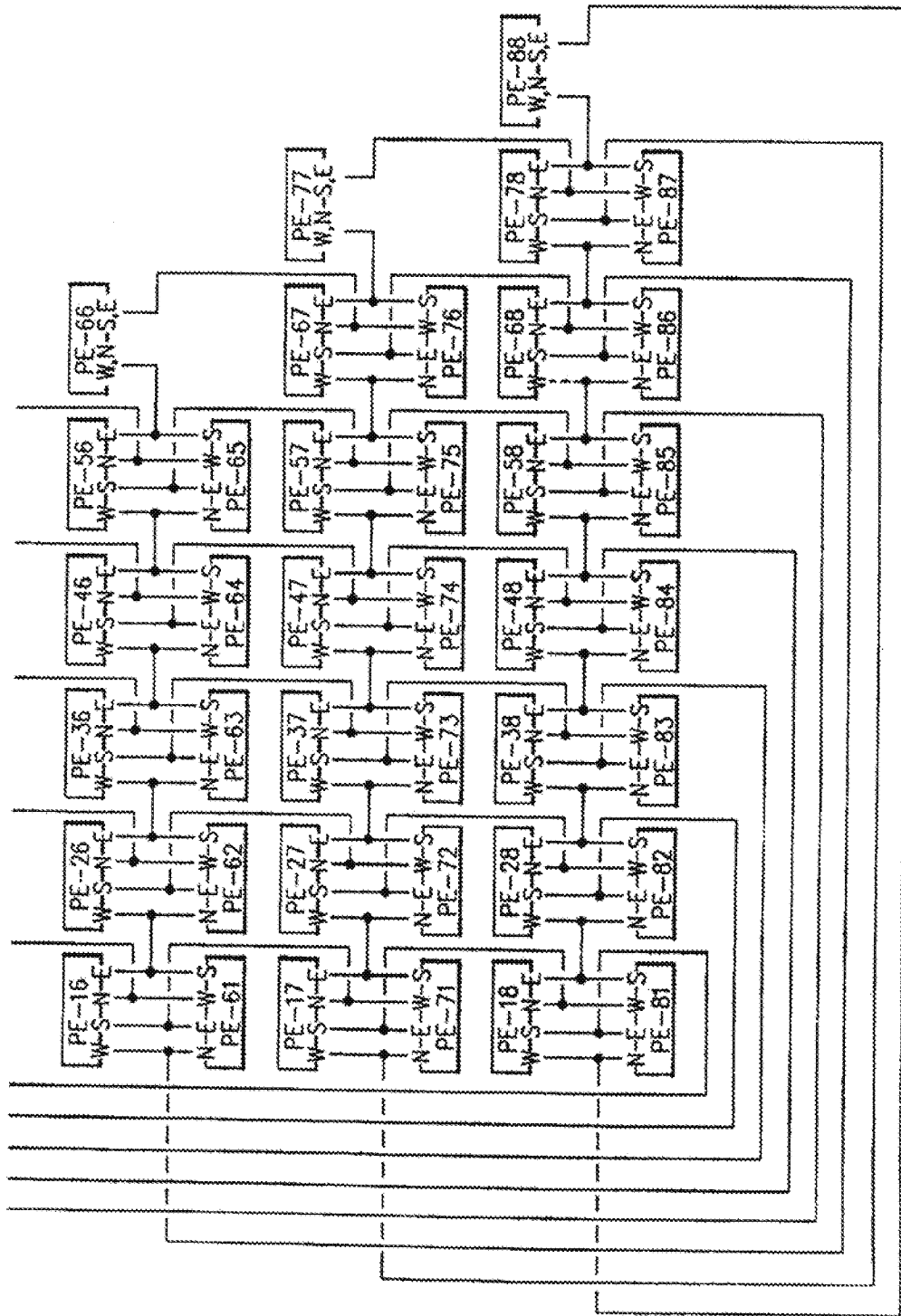


FIG. 3B

U.S. Patent

Jun. 11, 2002

Sheet 5 of 22

US 6,405,185 B1

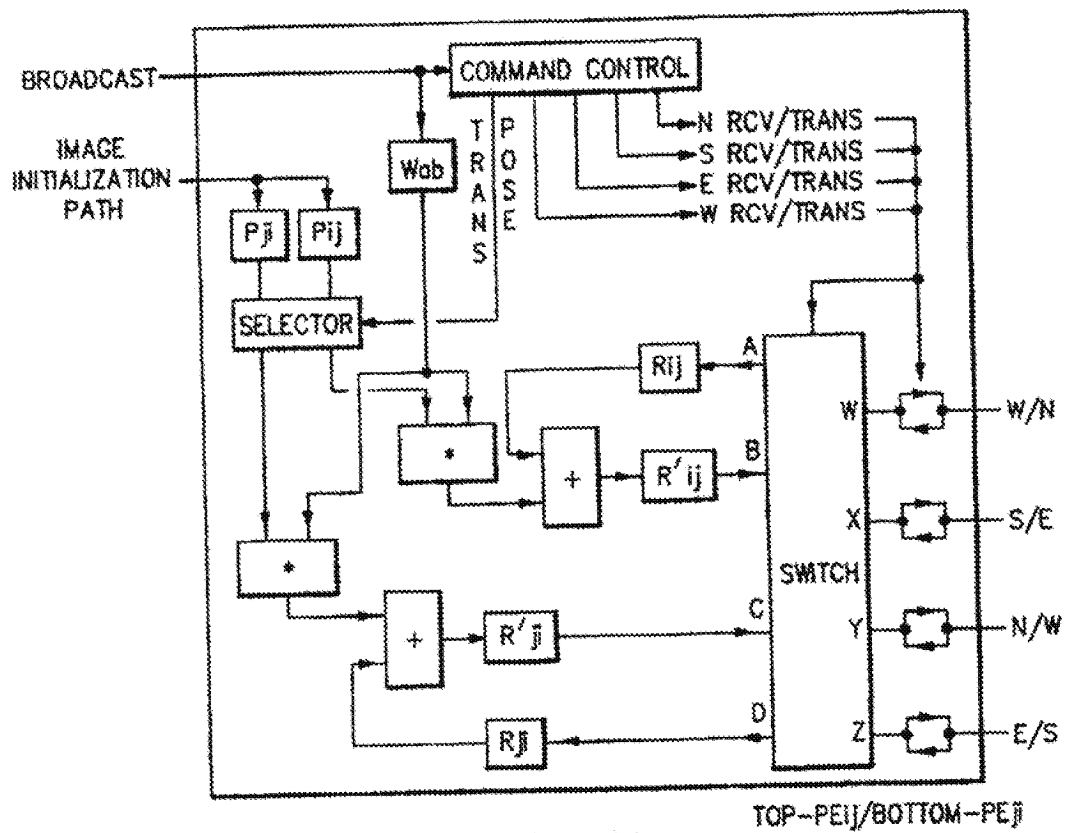


FIG. 4A

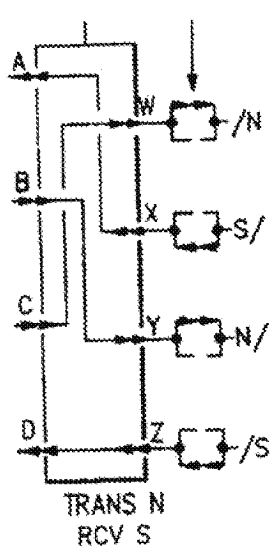


FIG. 4B

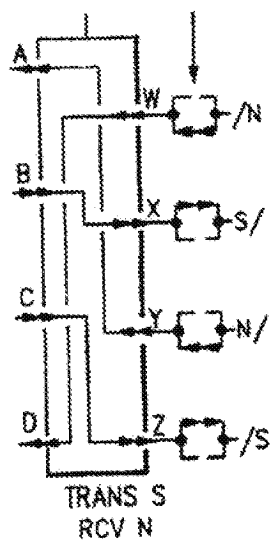


FIG. 4C

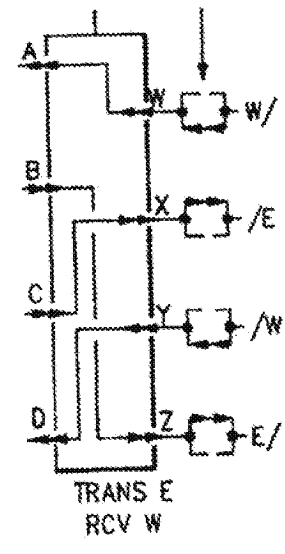


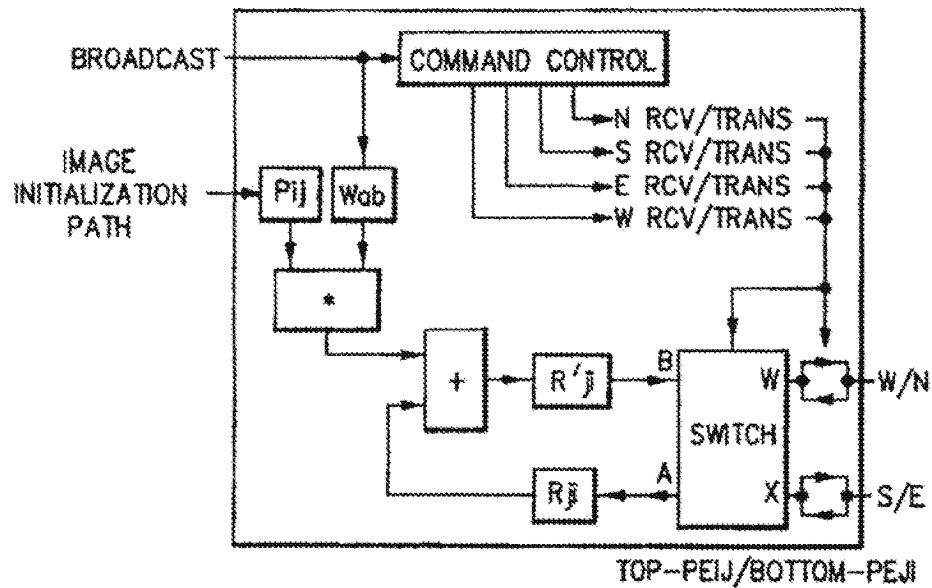
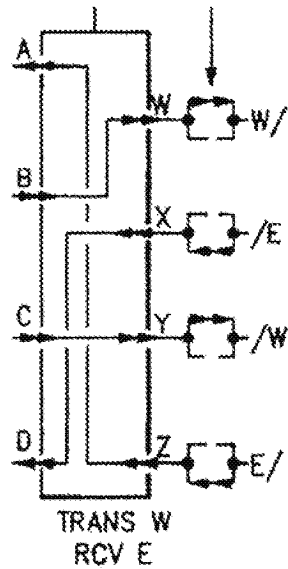
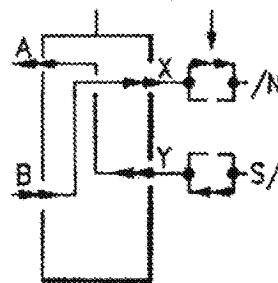
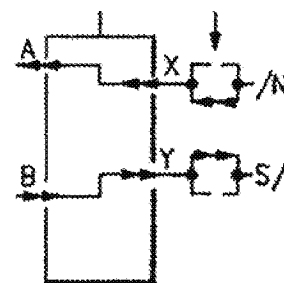
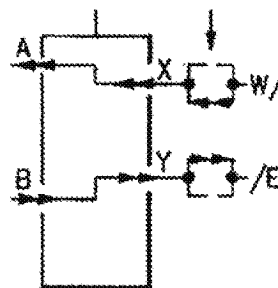
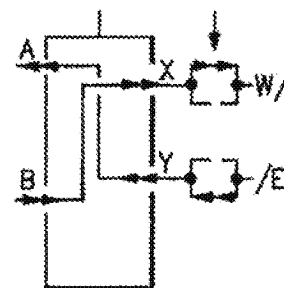
FIG. 4D

U.S. Patent

Jun. 11, 2002

Sheet 6 of 22

US 6,405,185 B1

**FIG. 4F****FIG. 4E****FIG. 4G****FIG. 4H****FIG. 4I****FIG. 4J**

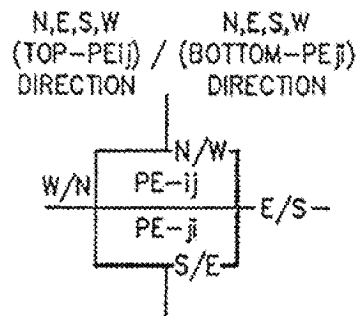
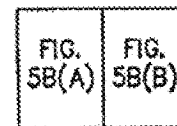
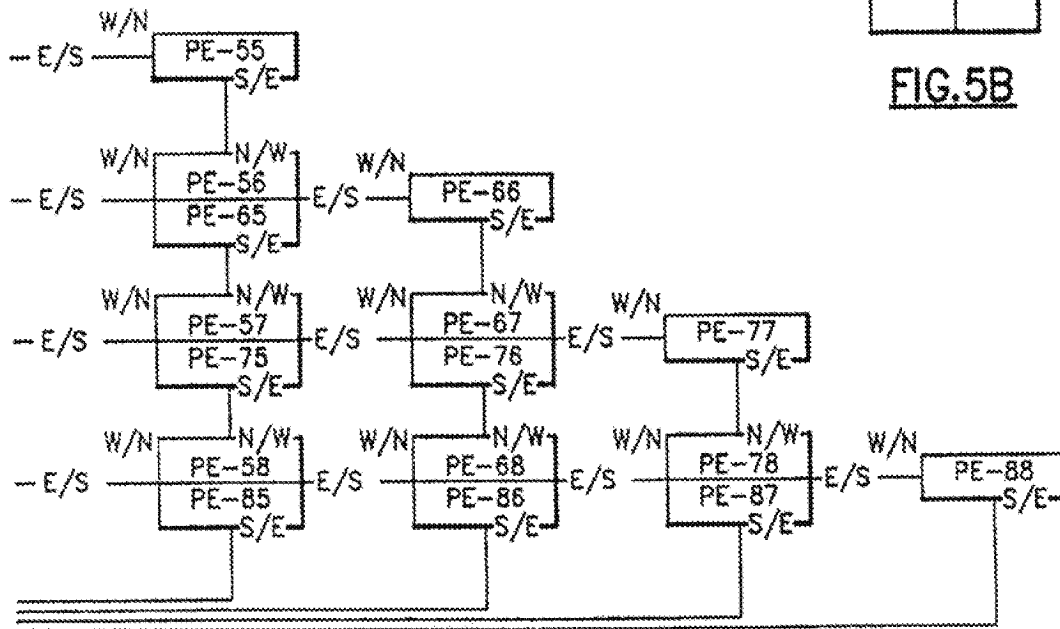


U.S. Patent

Jun. 11, 2002

Sheet 7 of 22

US 6,405,185 B1

FIG. 5AFIG. 5BFIG. 5B(B)

U.S. Patent

Jun. 11, 2002

Sheet 8 of 22

US 6,405,185 B1

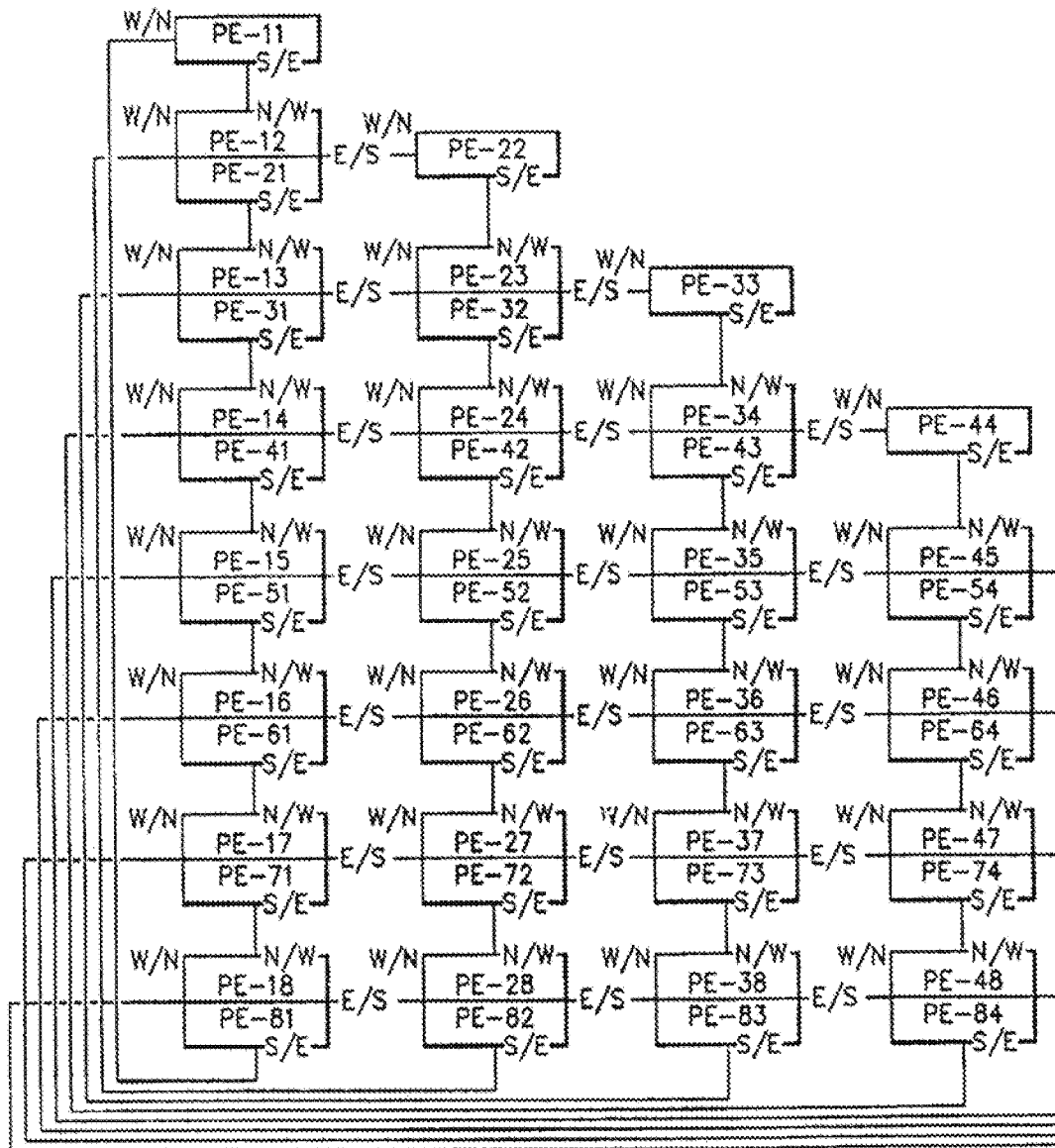


FIG. 5B(A)

U.S. Patent

Jun. 11, 2002

Sheet 9 of 22

US 6,405,185 B1

1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8

FIG. 7A

1								
2	2							
1								
3	3	3						
1	2							
4	4	4	4					
1	2	3						
5	5	5	5	5				
1	2	3	4					
6	6	6	6	6	6			
1	2	3	4	5				
7	7	7	7	7	7	7		
1	2	3	4	5	6			
8	8	8	8	8	8	8	8	8
1	2	3	4	5	6	7		

FIG. 7B

W13	W23	W33
W12	W22	W32
W11	W21	W31

FIG. 8

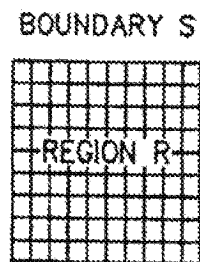


FIG. 13A

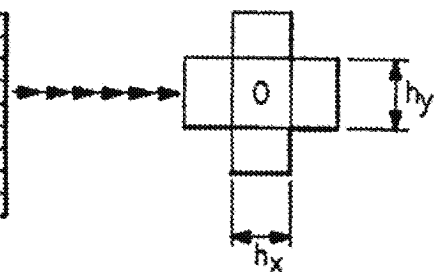


FIG. 13B

U.S. Patent

Jun. 11, 2002

Sheet 10 of 22

US 6,405,185 B1

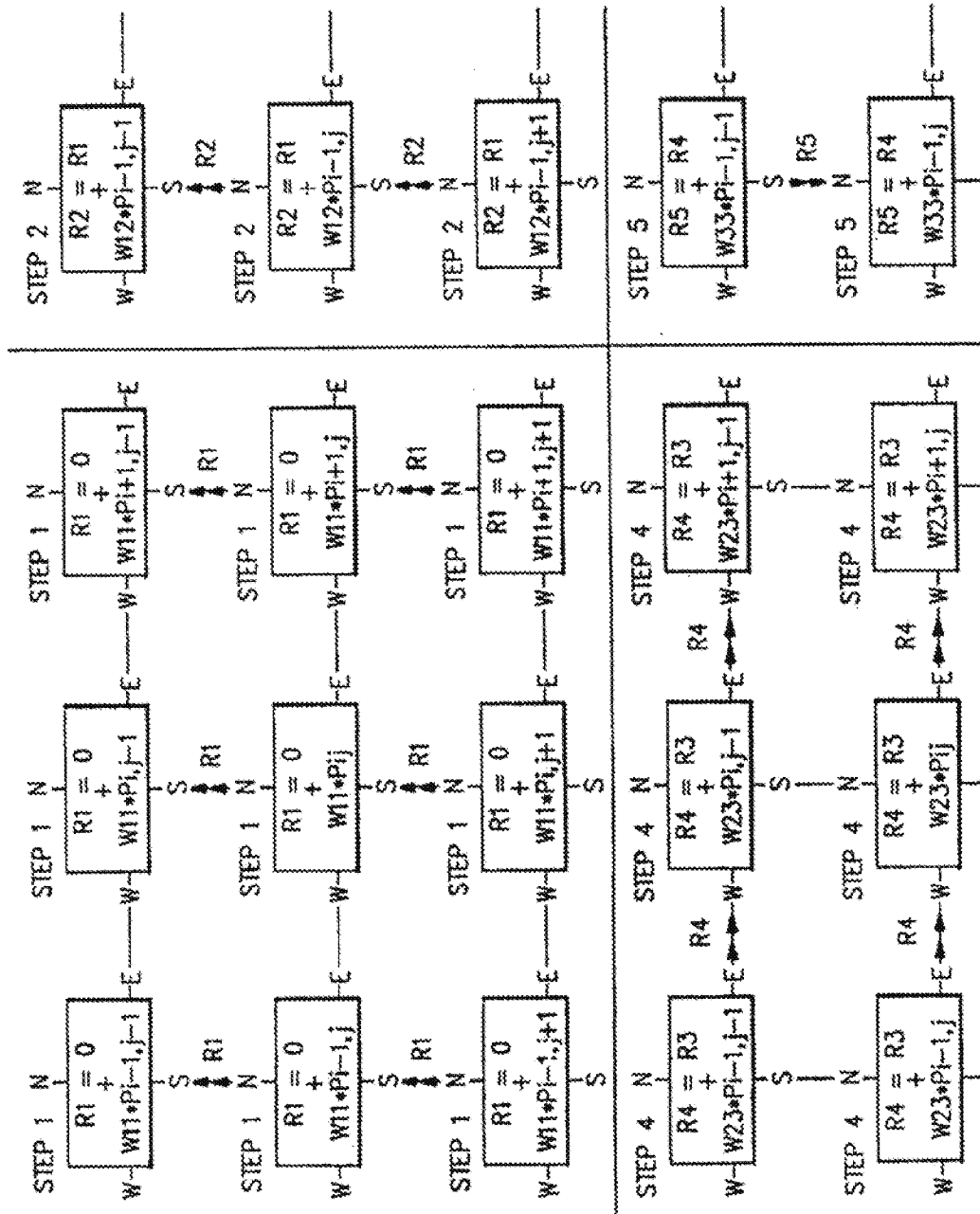


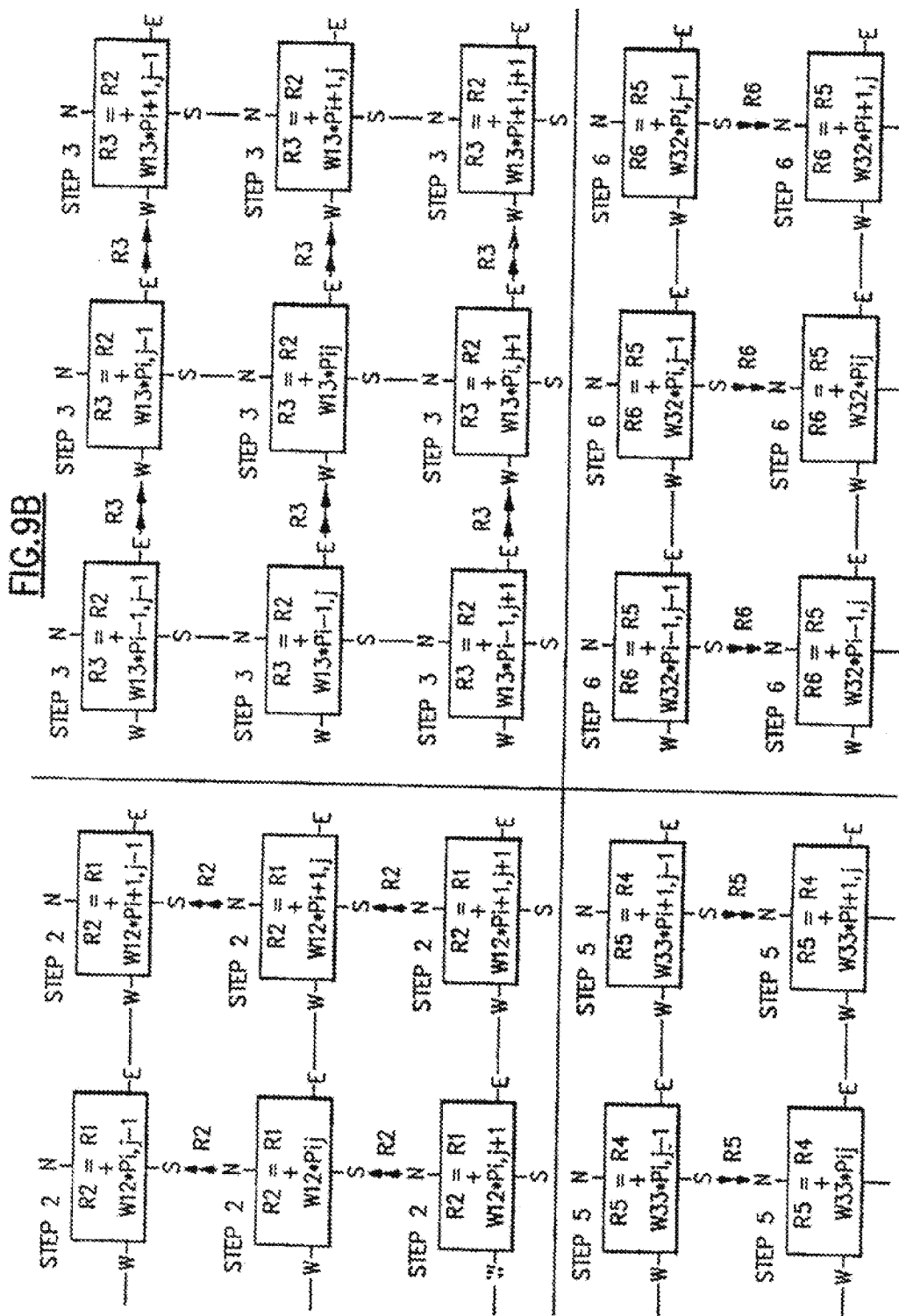
FIG. 9A

U.S. Patent

Jun. 11, 2002

Sheet 11 of 22

US 6,405,185 B1





U.S. Patent

Jun. 11, 2002

Sheet 12 of 22

US 6,405,185 B1

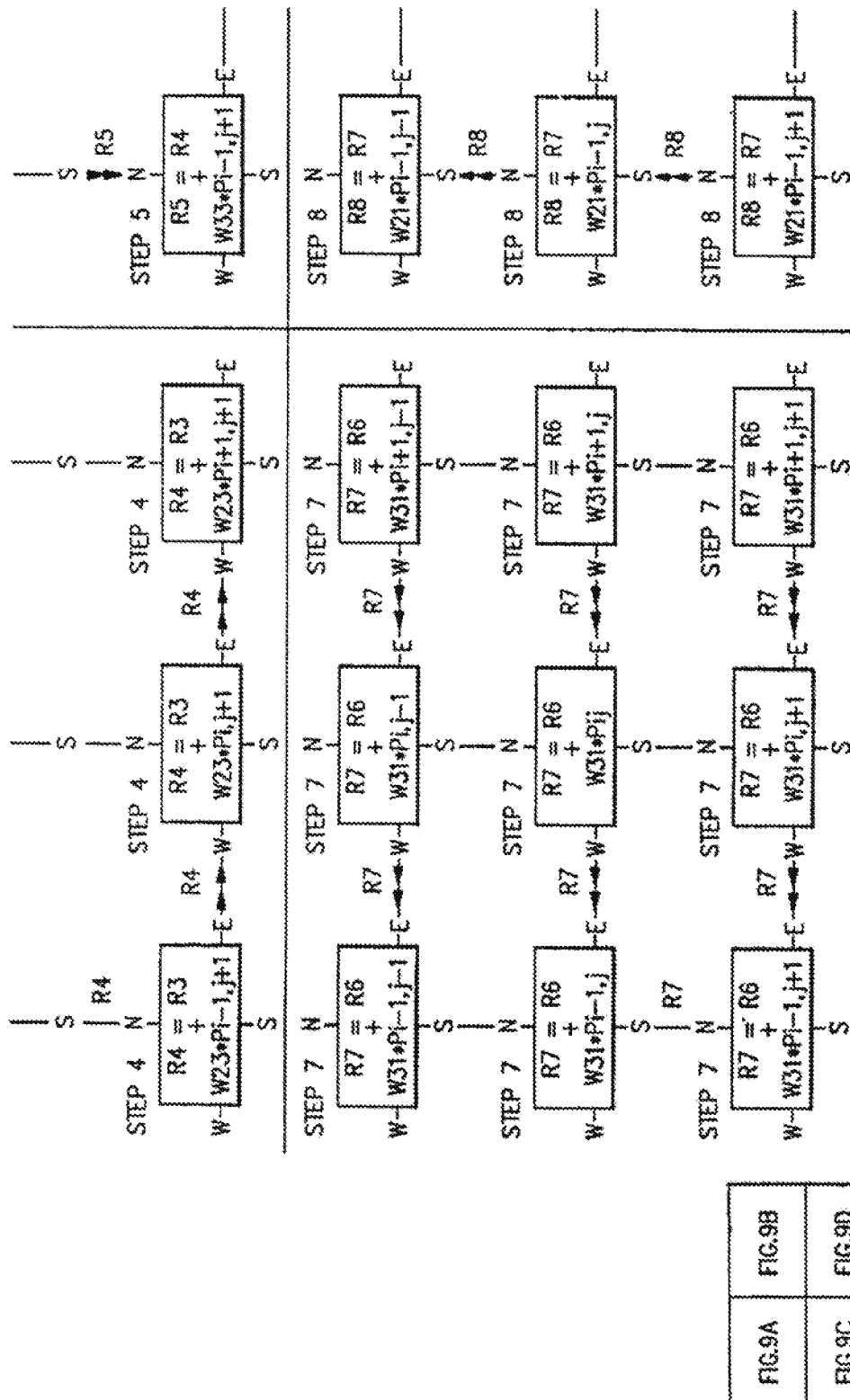


FIG. 9

FIG. 9C

U.S. Patent

Jun. 11, 2002

Sheet 13 of 22

US 6,405,185 B1

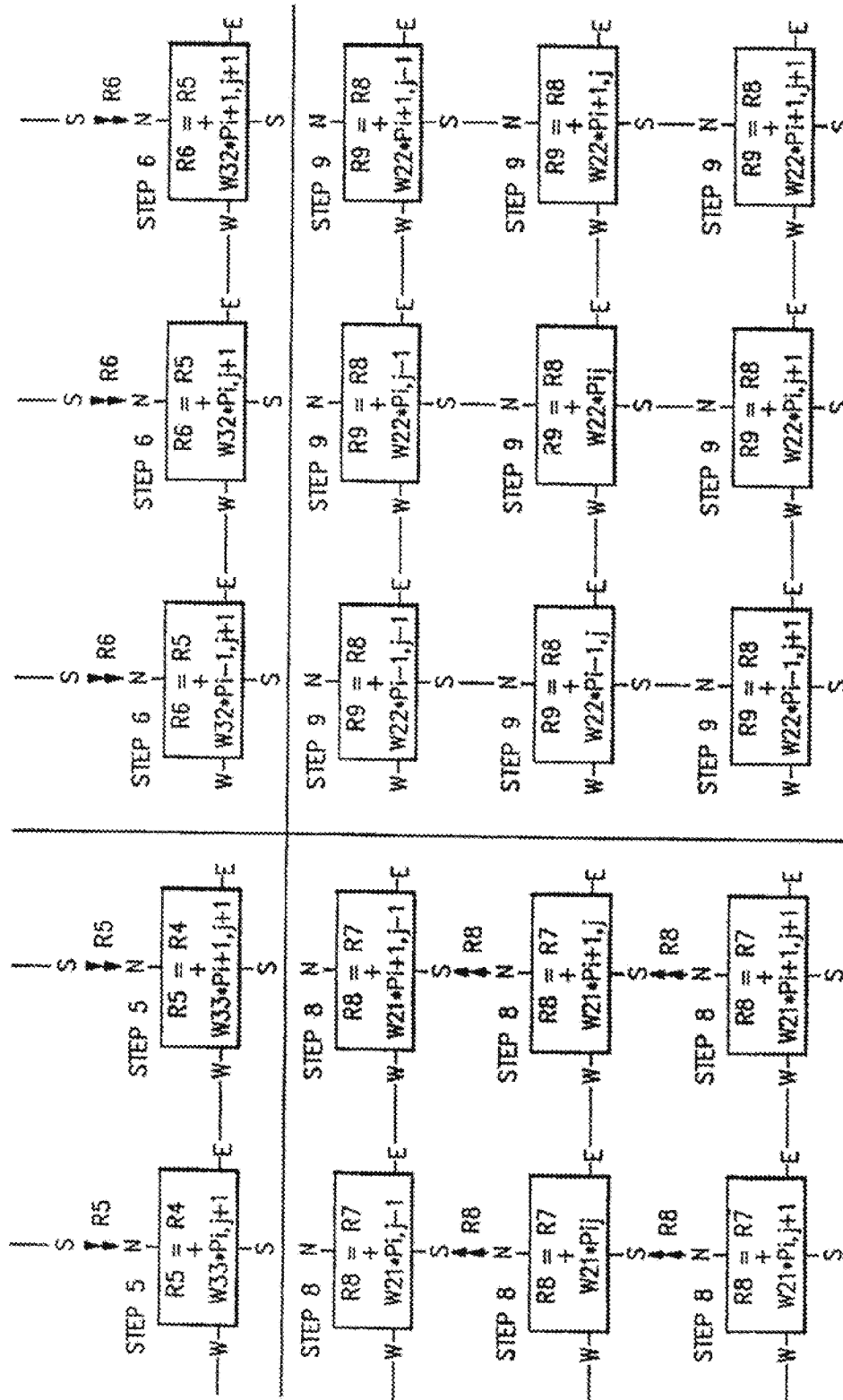


FIG. 9D

U.S. Patent

Jun. 11, 2002

Sheet 14 of 22

US 6,405,185 B1

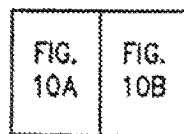
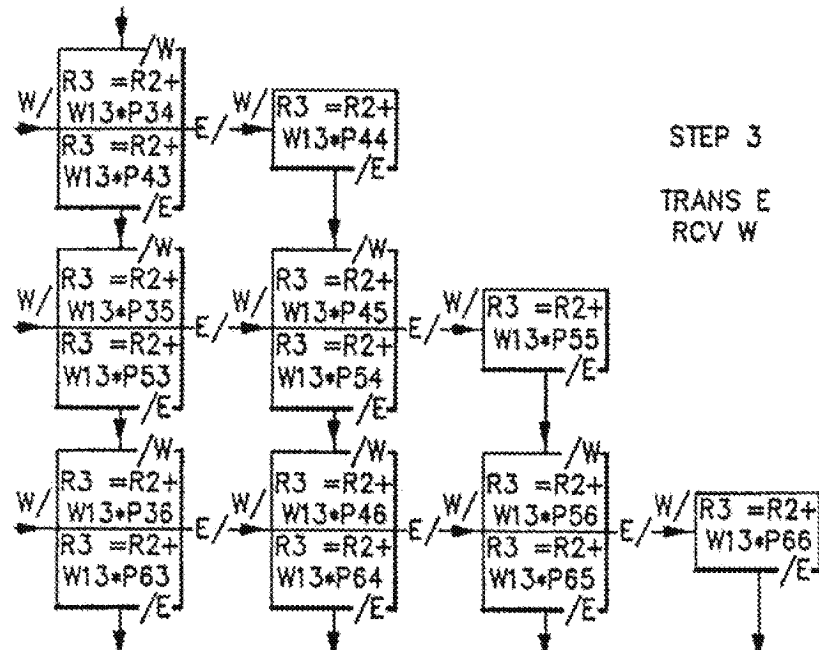
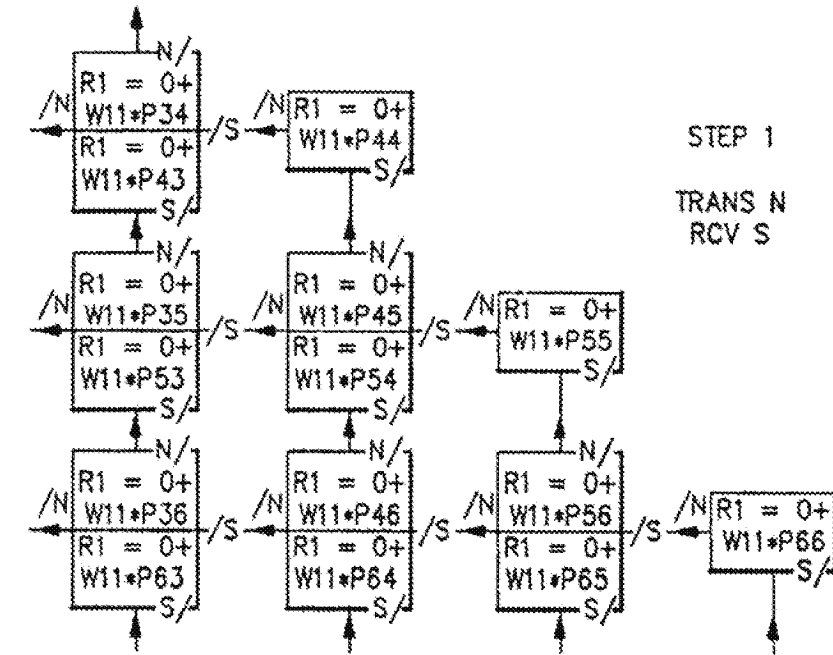


FIG.10

FIG.10A

U.S. Patent

Jun. 11, 2002

Sheet 15 of 22

US 6,405,185 B1

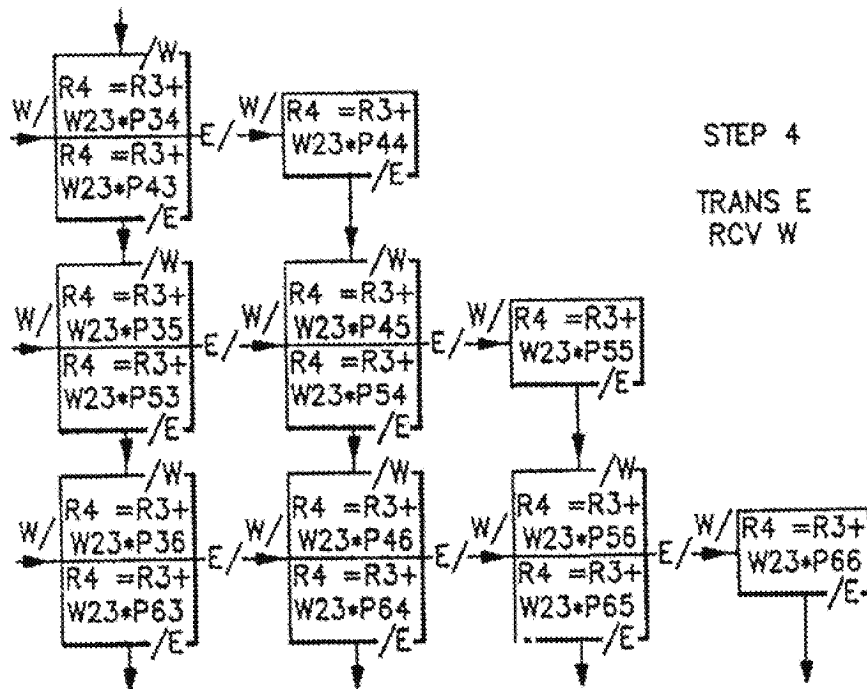
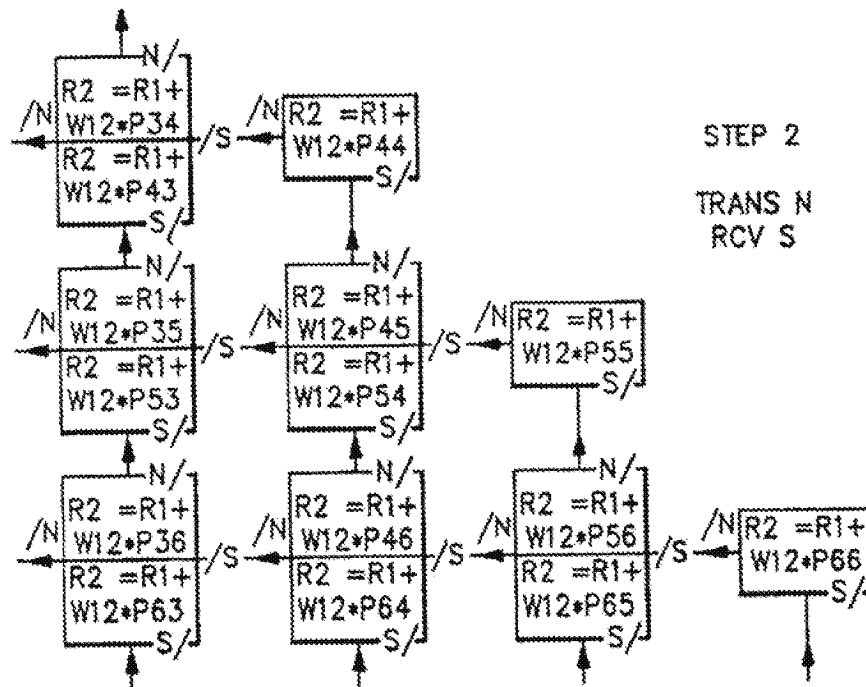


FIG.10B

U.S. Patent

Jun. 11, 2002

Sheet 16 of 22

US 6,405,185 B1

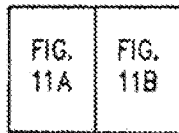
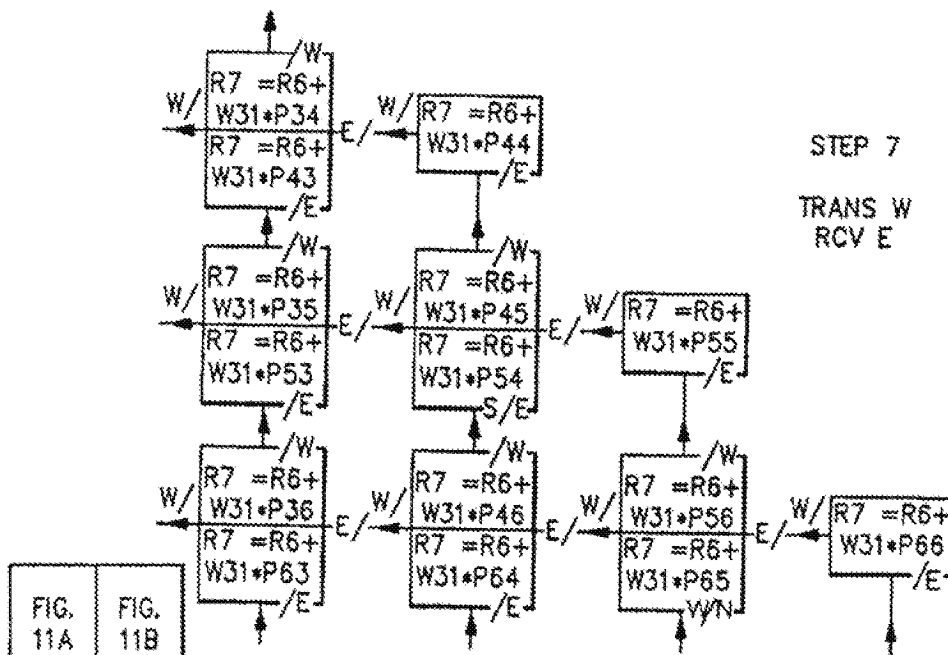
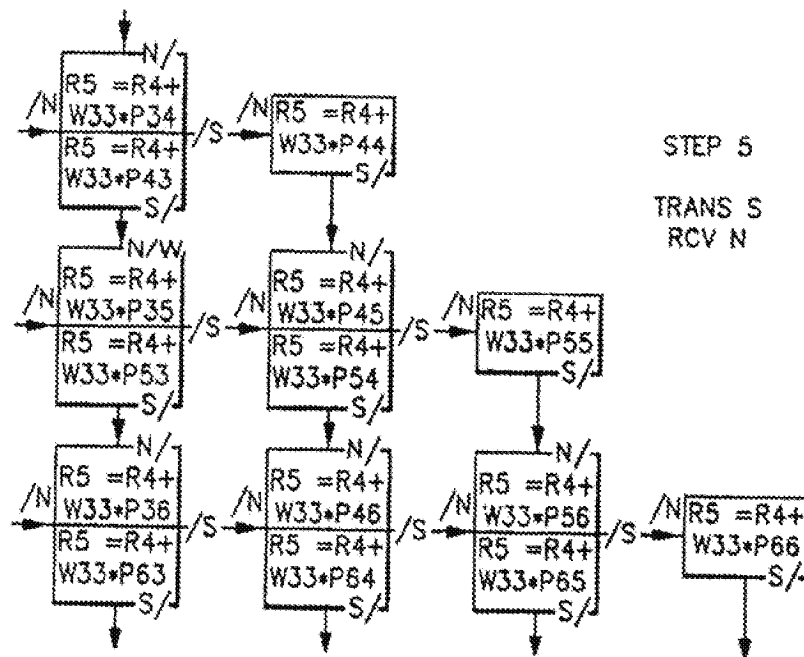


FIG.11

FIG.11A



U.S. Patent

Jun. 11, 2002

Sheet 17 of 22

US 6,405,185 B1

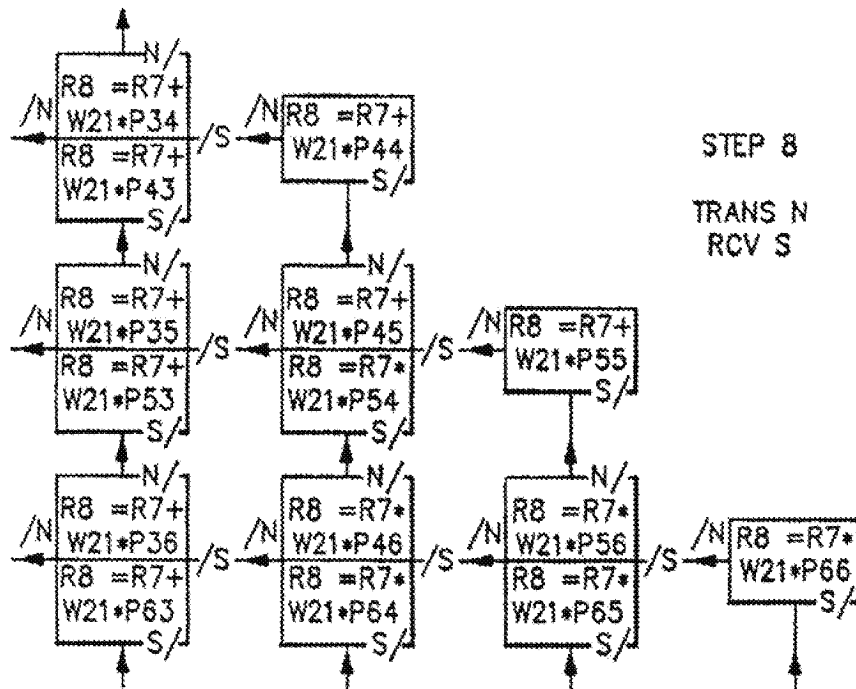
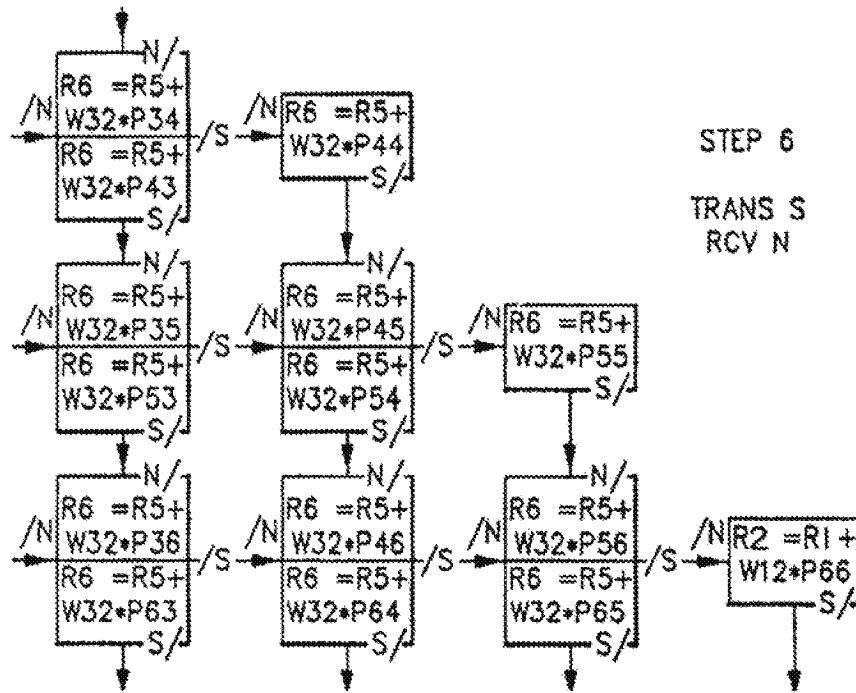


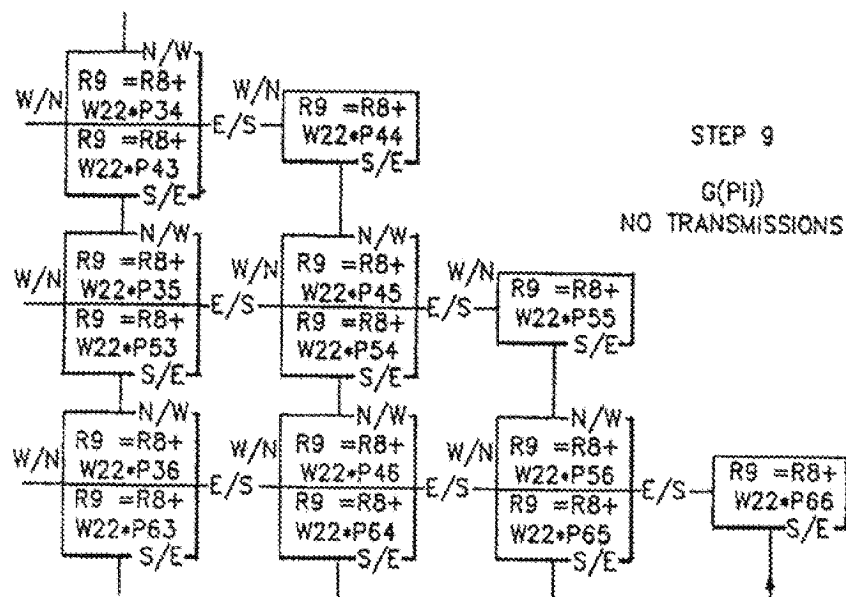
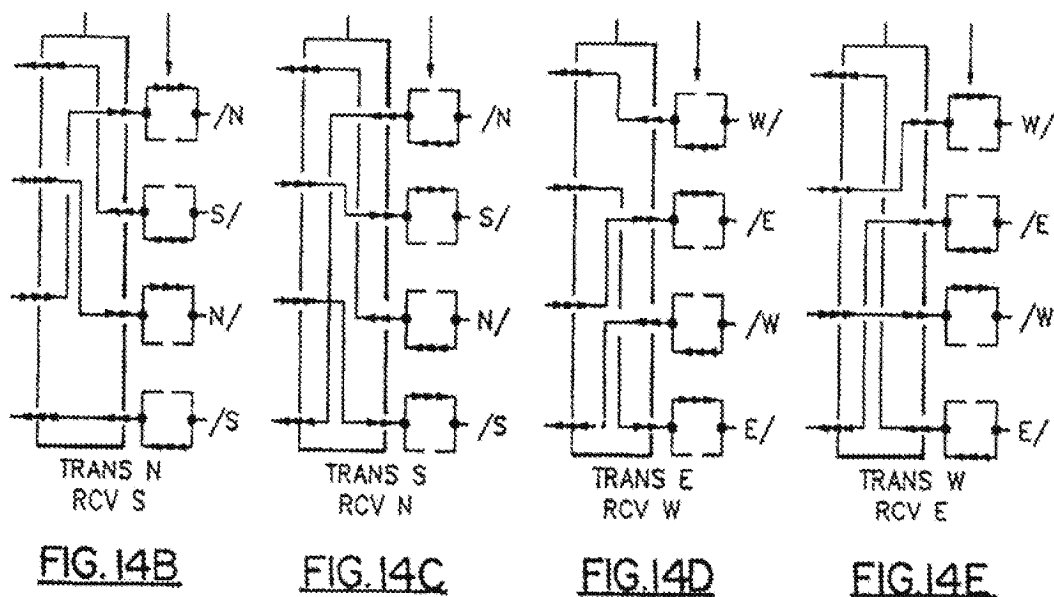
FIG.11B

U.S. Patent

Jun. 11, 2002

Sheet 18 of 22

US 6,405,185 B1

**FIG. 12**

U.S. Patent

Jun. 11, 2002

Sheet 19 of 22

US 6,405,185 B1

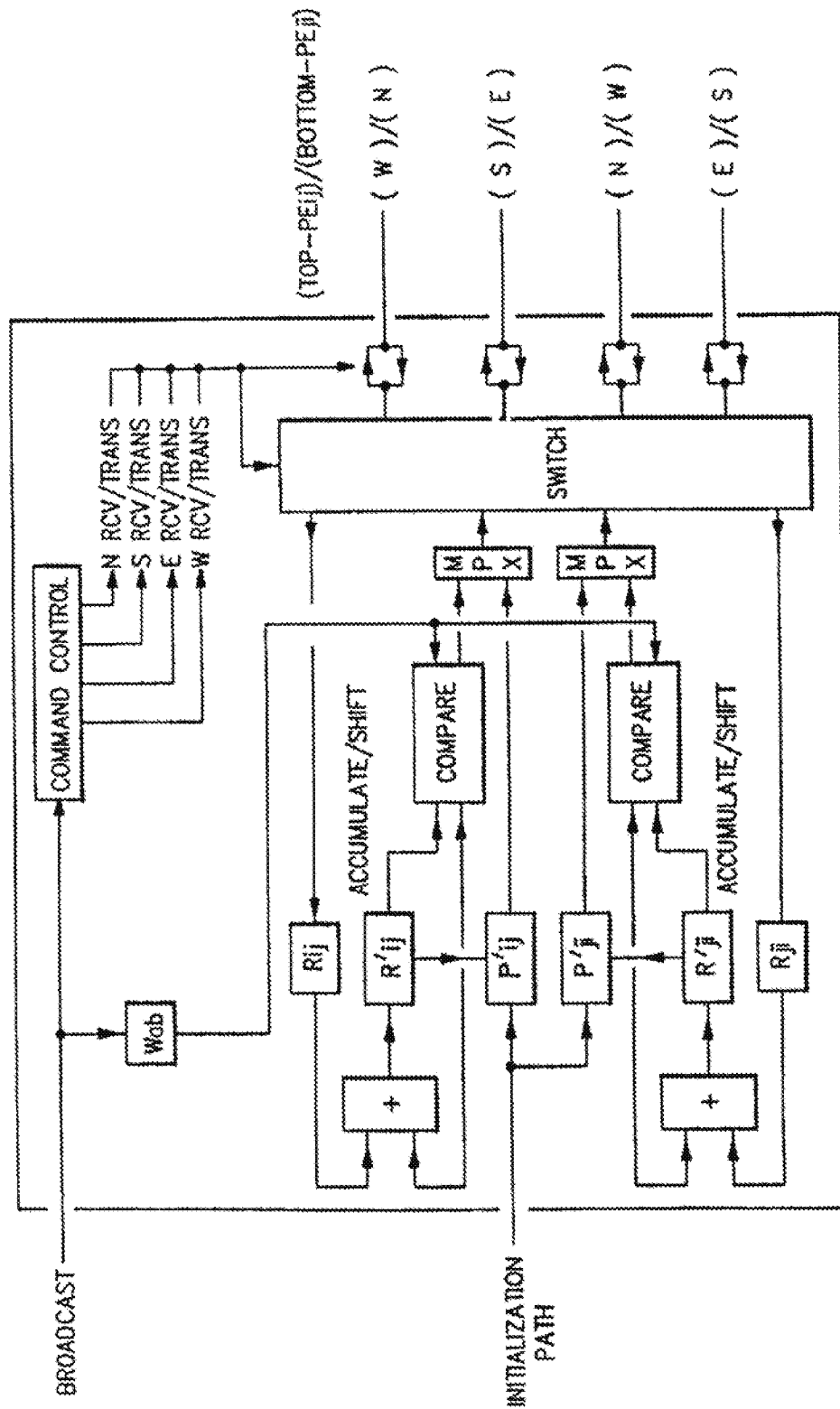


FIG. 14A

U.S. Patent

Jun. 11, 2002

Sheet 20 of 22

US 6,405,185 B1

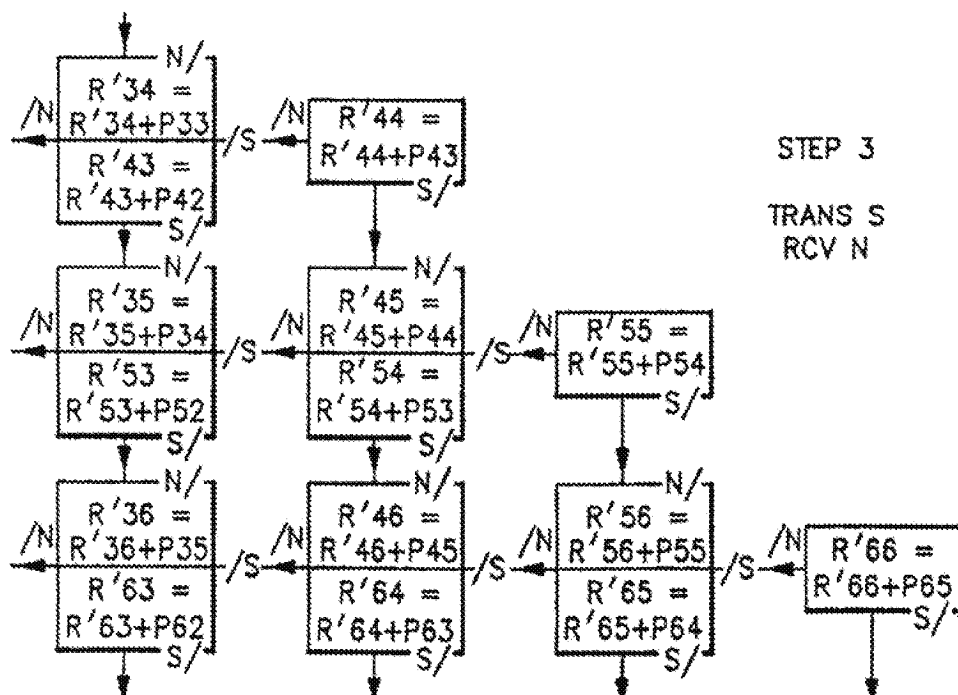
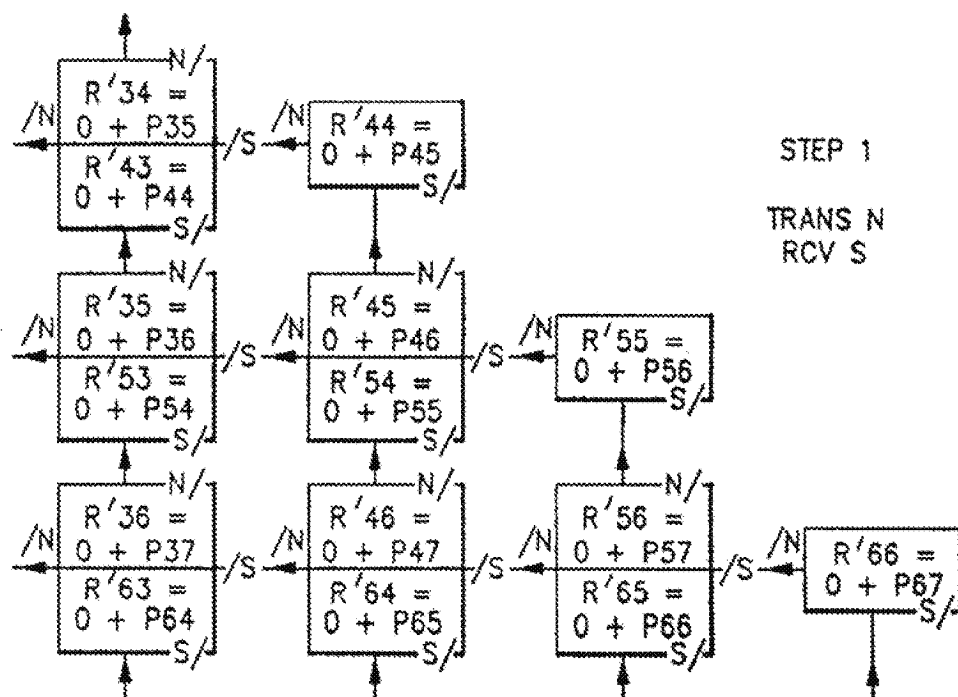


FIG.15A

U.S. Patent

Jun. 11, 2002

Sheet 21 of 22

US 6,405,185 B1

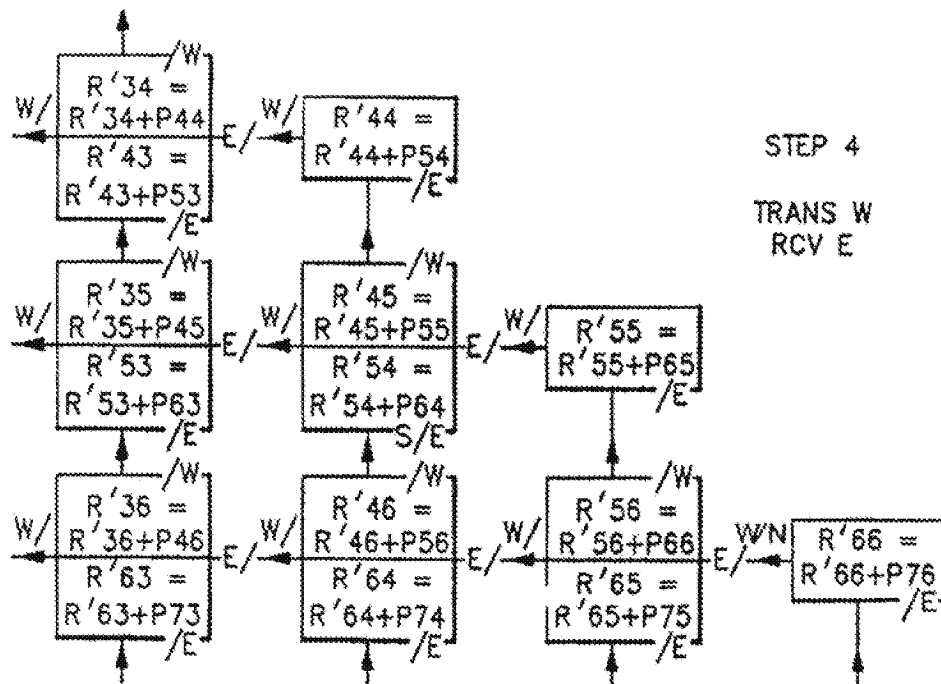
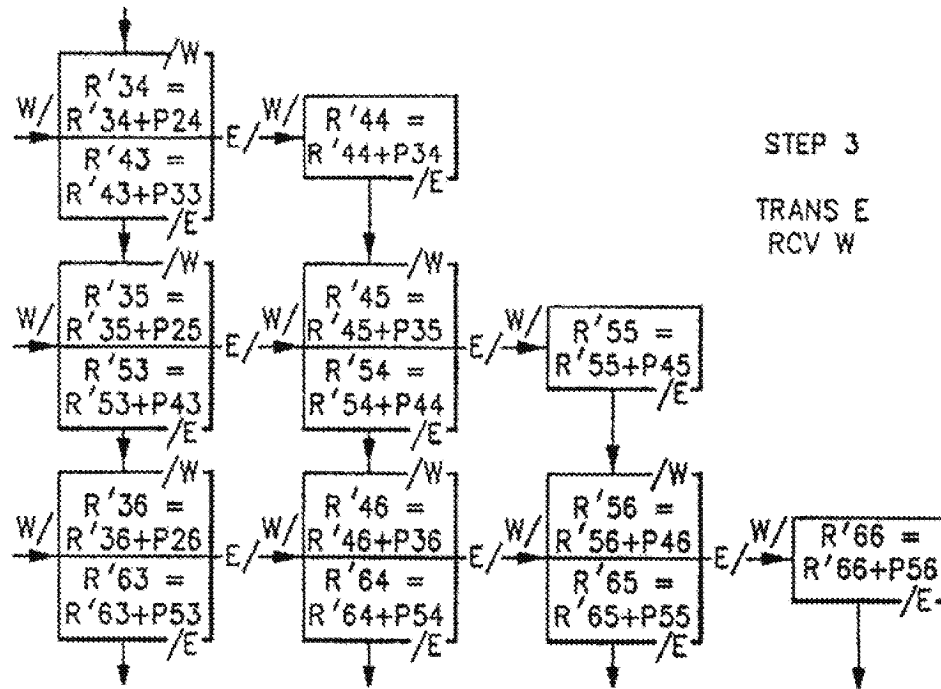


FIG.15B

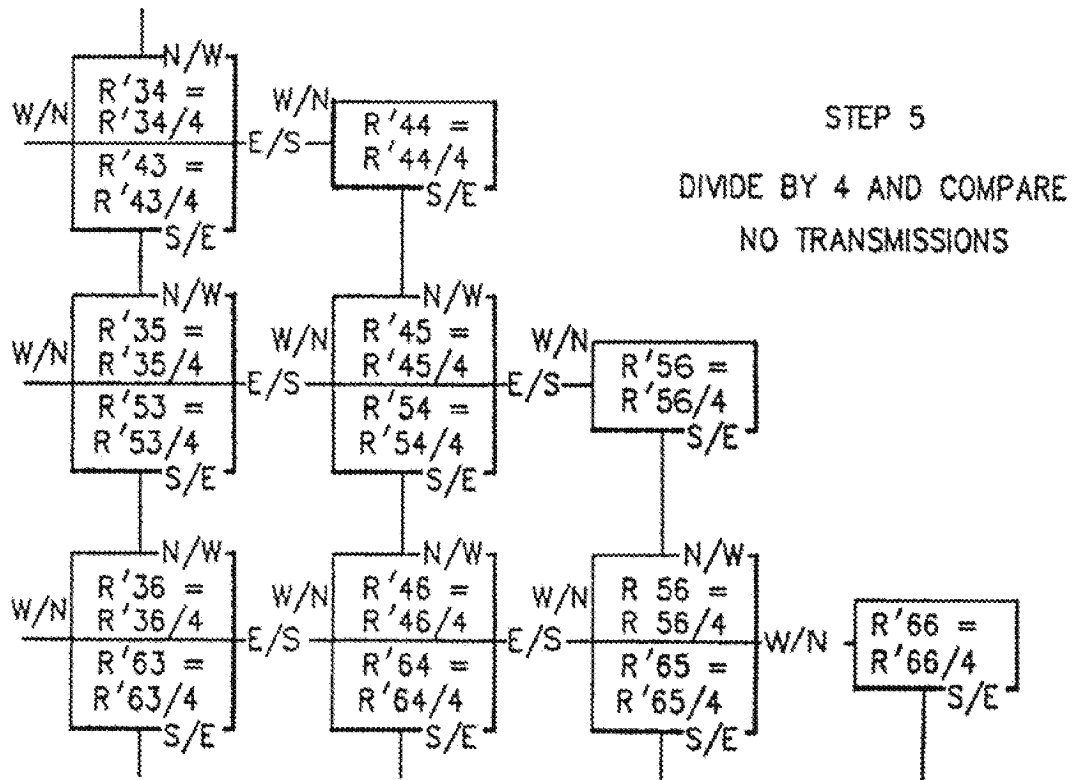
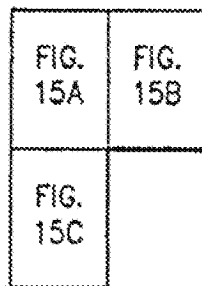


U.S. Patent

Jun. 11, 2002

Sheet 22 of 22

US 6,405,185 B1

FIG.15CFIG.15

US 6,405,185 B1

1

## MASSIVELY PARALLEL ARRAY PROCESSOR

This application is a continuation of application Ser. No. 07/864,112 filed Apr. 6, 1992, now abandoned.

### FIELD OF THE INVENTION

The invention relates to computers and particularly to parallel array processors.

### CROSS REFERENCE TO RELATED APPLICATIONS

The present U.S. patent application claims priority as a continuation-in-part application and is related to the following applications:

U.S. Ser. No. 07/526,866 filed May 22, 1990, of S. Vassiliadis et al, entitled: Orthogonal Row-Column Neural Processor (now U.S. Pat. No. 5,065,339, issued Nov. 12, 1991); and

U.S. Ser. No. 07/740,355 filed Aug. 5, 1991, of S. Vassiliadis et al, entitled: Scalable Neural Array Processor, issued as U.S. Pat. No. 5,146,543; and,

U.S. Ser. No. 07/740,556 filed Aug. 5, 1991, of S. Vassiliadis et al, entitled: Adder Tree for a Neural Array Processor, issued as U.S. Pat. No. 5,146,420 and,

U.S. Ser. No. 07/740,568 filed Aug. 5, 1991, of S. Vassiliadis et al, entitled: Apparatus and Method for Neural Processor, abandoned in favor of U.S. Ser. No. 08/000,915, filed Jan. 6, 1993, issued as U.S. Pat. No. 5,251,287 and,

U.S. Ser. No. 07/740,266 filed Aug. 5, 1991, of S. Vassiliadis et al, entitled: Scalable Neural Array Processor and Method, issued as U.S. Pat. No. 5,148,515 and

U.S. Ser. No. 07/682,786 filed Apr. 8, 1991, of G. G. Pechanek et al, entitled: Triangular Scalable Neural Array Processor, abandoned in favor of continuation application U.S. Ser. NO. 08/231,853, filed Apr. 22, 1994, (now co-pending).

These applications and the present continuation-in-part application are owned by one and the same assignee, namely, International Business Machines Corporation of Armonk, N.Y.

The descriptions set forth in these above applications are hereby incorporated into the present application.

### GLOSSARY OF TERMS

#### ALU

ALU is the arithmetic logic unit portion of a processor.

#### Array

Array refers to an arrangement of elements in one or more dimensions.

Array processors are computers which have many functional units or PEs arranged and interconnected to process in an array. Massively parallel machines use array processors for parallel processing of data arrays by array processing elements or array elements. An array can include an ordered set of data items (array element) which in languages like Fortran are identified by a single name, and in other languages such a name of an ordered set of data items refers to an ordered collection or set of data elements, all of which have identical attributes. An program array has dimensions specified, generally by a number or dimension attribute. The declarator of the array may also specify the size of each dimension of the array in some languages. In some languages, an array is an arrangement of elements in a table. In a hardware sense, an array is a collection of structures

2

(functional elements) which are generally identical in a massively parallel architecture. Array elements in data parallel computing are elements which can be assigned operations, and when parallel can each independently and in parallel execute the operations required. Generally arrays may be thought of as grids of processing elements. Sections of the array may be assigned sectional data, so that sectional data can be moved around in a regular grid pattern. However, data can be indexed or assigned to an arbitrary location in an array.

#### Functional unit

A functional unit is an entity of hardware, software, or both, capable of accomplishing a purpose.

#### MIMD

A processor array architecture wherein each processor in the array has its own instruction stream, thus Multiple Instruction stream, to execute Multiple Data streams located one per processing element.

#### Module

A module is a program unit that is discrete and identifiable, or a functional unit of hardware designed for use with other components.

#### PE

PE is used for processing element. We use the term PE to refer to a single processor, which has interconnected allocated memory and I/O capable system element or unit that forms one of our parallel array processing elements. As the result of wiring, in our system, symmetric replicatable elements, are wired together for sharing interconnection paths.

#### SIMD

A processor array architecture wherein all processors in the array are commanded from a Single Instruction stream, to execute Multiple Data streams located one per processing element.

### REFERENCES USED IN THE DISCUSSION OF THE INVENTION

During the detailed description which follows the following works will be referenced as an aid for the reader. These additional references are:

1. R. J. Gove, W. Lee, Y. Kim, and T. Alexander, "Image Computing Requirements for the 1990s: from Multimedia to Medicine," *Proceedings of the SPIE Vol. 1444—Image Capture, Formatting, and Display*, pp. 318-333, 1991.
2. R. Cypher and J. L. C. Sanz, "SIMD Architectures and Algorithms for Image Processing and Computer Vision," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, No. 12, pp. 2158-2174, December 1989.
3. K. E. Batchner, "Design of a Massively Parallel Processor," *IEEE Transactions on Computers* Vol. C-29, No. 9, pp. 836-840, September 1980.
4. L. Uhr, *Multi-Computer Architectures for Artificial Intelligence*, New York, N.Y.: John Wiley & Sons, chap. 8, p.97, 1987.
5. S.-Y. Lee and J. K. Aggarwal, "Parallel 2-D Convolution on a Mesh Connected Array Processor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 4, pp. 590-594, July 1987.
6. E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for Testability," *Proc. 14th Design Automation Conference*, IEEE, 1977.
7. D. M. Young and D. R. Kincaid, "A Tutorial on Finite Difference Methods and Ordering of Mesh Points,"

US 6,405,185 B1

3

*Proceedings of the Fall Joint Computer Conference*, pp. 556-559, Dallas, Tex.: IEEE Press, November 1986.

8. E. Kreyszig, *Advanced Engineering Mathematics*, New York, N.Y.: John Wiley & Sons, chap. 9.7, pp. 510-512, 1968.

9. U.S. Ser. No. 07/799,602, filed Nov. 27, 1991, by H. Olmowich, entitled: "Multi-Media Serial Line Switching Adapter for Parallel Networks and Heterogeneous and Homologous Computer Systems", systems which allow dynamic switching between MIMD, SIMD, and SISD.

10. U.S. Ser. No. 07/798,788, filed Nov. 27, 1991, by P. M. Kogge, entitled: "Dynamic Multi-mode Parallel Processor Array Architecture".

These additional references are incorporated by reference.

### BACKGROUND OF THE INVENTION

As background for our invention, the processing of visual information can be considered to consist of three different processing domains: image processing, pattern recognition, and computer graphics. The merger of image processing, pattern recognition and computer graphics is referred to as image computing and represents a capability required by the multimedia workstations of the future. "Multimedia refers to a technique that presents information in more than one way, such as via images, graphics, video, audio, and text, in order to enhance the comprehensibility of the information and to improve human-computer interaction" (See Additional Reference 1).

In the never ending quest for faster computers, engineers are linking hundreds, and even thousands of low cost microprocessors together in parallel to create super super-computers that divide in order to conquer complex problems that stump today's machines. Such machines are called massively parallel. Multiple computers operating in parallel have existed for decades.

Early parallel machines included the ILLIAC which was started in the 1960s. Other multiple processors include (see a partial summary in U.S. Pat. No. 4,975,834 issued Dec. 4, 1990 to Xu et al) the Cedar, Sigma-1, the Butterfly and the Monarch, the Intel ipsc, The Connection Machines, the Caltech COSMIC, the N Cube, IBM's RP3, IBM's GFT, the NYU Ultra Computer, the Intel Delta and Touchstone.

Large multiple processors beginning with ILLIAC have been considered supercomputers. Supercomputers with greatest commercial success have been based upon multiple vector processors, represented by the Cray Research Y-MP systems, the IBM 3090, and other manufacturer's machines including those of Amdahl, Hitachi, Fujitsu, and NEC.

Massively Parallel Processors (MPPs) are now thought of as capable of becoming supercomputers. These computer systems aggregate a large number of microprocessors with an interconnection network and program them to operate in parallel. There have been two modes of operation of these computers. Some of these machines have been MIMD mode machines. Some of these machines have been SIMD mode machines. Perhaps the most commercially acclaimed of these machines has been the Connection Machines series 1 and 2 of Thinking Machines, Inc. These have been essentially SIMD machines. Many of the massively parallel machines have used microprocessors interconnected in parallel to obtain their concurrency or parallel operations capability. Intel microprocessors like i860 have been used by Intel and others. N Cube has made such machines with Intel 386 microprocessors. Other machines have been built with

4

what is called the "transputer" chip. Inmos Transputer IMS T800 is an example. The Inmos Transputer T800 is a 32 bit device with an integral high speed floating point processor.

As an example of the kind of systems that are built, several Inmos Transputer T800 chips each would have 32 communication link inputs and 32 link outputs. Each chip would have a single processor, a small amount of memory, and communication links to the local memory and to an external interface. In addition, in order to build up the system communication link adaptors like IMS C011 and C012 would be connected. In addition switches, like a IMS C004 would be provided to provide, say, a crossbar switch between the 32 link inputs and 32 link outputs to provide point to point connection between additional transputer chips. In addition, there will be special circuitry and interface chips for transputers adapting them to be used for a special purpose tailored to the requirements of a specific device, a graphics or disk controller. The Inmos IMS M212 is a 16 bit process, with on chip memory and communication links. It contains hardware and logic to control disk drives and can be used as a programmable disk controller or as a general purpose interface. In order to use the concurrency (parallel operations) Inmos developed a special language, Occam, for the transputer. Programmers have to describe the network of transputers directly in an Occam program.

Some of these massively parallel machines use parallel processor arrays of processor chips which are interconnected with different topologies. The transputer provides a crossbar network with the addition of IMS C004 chips. Some other systems use a hypercube connection. Others use a bus or mesh to connect the microprocessors and there associated circuitry. Some have been interconnected by circuit switch processors that use switches as processor addressable networks. Generally, as with the 14 RISC/6000s which were interconnected last fall at Lawrence Livermore by wiring the machines together, the processor addressable networks have been considered as coarse-grained multiprocessors.

Some very large machines are being built by Intel and nCube and others to attack what are called "grand challenges" in data processing. However, these computers are very expensive. Recent projected costs are in the order of \$30,000,000.00 to \$75,000,000.00 (Tera Computer) for computers whose development has been funded by the U.S. Government to attack the "grand challenges". These "grand challenges" would include such problems as climate modeling, fluid turbulence, pollution dispersion, mapping of the human genome and ocean circulation, quantum chromodynamics, semiconductor and supercomputer modeling, combustion systems, vision and cognition.

### Problems Addressed by our Oracle Machine

It is a problem for massively parallel array processors to attack adequately the image computing problems which exist. One particular algorithm used in image processing is convolution, which replaces each image pixel value with a weighted sum of the pixels in a defined surrounding area or window of pixels. A  $M \times M$  square convolution window consists of a set of  $M \times M$  weights, each corresponding to the associated pixels located in the window (Additional Reference 2). For an  $N$  by  $N$  array of pixels, the convolution algorithm requires  $M^2 N^2$  multiplication operations. Assuming an  $N$  of 1024 and a  $M$  of 3 a single image frame convolution would take 9 million multiplications and sum of product calculations per convolution and if the processing is on video data occurring at a rate of 30 frames per second then 270 million multiplications sum of product calculations



US 6,405,185 B1

5

per second would be required. For a uniprocessor to process this data, where each convolution window weight value must be fetched separately, with the multiply and add treated as separate operations, and followed by a write of the weighted average pixel result, the convolution would consist of 27 separate operations per pixel (9 reads, 9 multiplies, 8 adds, and 1 write) resulting in 27 millionx30 operations per second or 810 million operations per second (Additional Reference 1). Due to the high computational load, special purpose processors have been proposed to off load the image processing task from the system processor and to provide the adequate throughput required for image computing. One of these special purpose processors is the nearest neighbor mesh connected computer (See Additional References 2, 3, and 4-pp. 97) where multiple Processor Elements (PEs) are connected to their north, south, east west neighbor PEs and all PEs are operated in a synchronous Single Instruction Multiple Data (SIMD) fashion. It is assumed that a PE can communicate with any of its neighboring PEs but only one neighbor PE at a time. For example, each PE can communicate to their east neighbor PE, in one communication cycle. It is also assumed that a broadcast mechanism is present such that data and instructions can be communicated simultaneously to all PEs in one broadcast communication period. Bit serial interfaces are typical, as they were present in the Thinking Machines CM-1 family.

As is thus recognized, what is needed is a PE which can improve image computing, improve speed, and be adaptable to be replicated as part of a parallel array processor in a massively parallel environment. There is a need to improve the system apparatus for use in solving differential equations. We think a new kind of PE is needed for this problem. Creation of a new PE and massively parallel computing system apparatus built with new thought will improve the complex processes which need to be handled in the multimedia image computer field, and still be able to process general purpose applications.

#### SUMMARY OF THE INVENTION

The improvements which we have made result in a new machine apparatus. We call the machine which implements our invention the Oracle machine and we will describe it below. Our present invention relates to the apparatus which enables making a massively parallel computing system. We present a new PE and related organizations of computer systems which can be employed in a parallel array computer system or massively parallel array processor.

We provide a massively parallel computer system for multi-media and general purpose applications, including the use of a finite difference method of solving differential equations. Our processor is a triangular processor array structure. Our processor array structure has single and dual processing elements that contain instruction and data storage units, receive instructions and data, and execute instructions and a processor interconnection organization and a way to support data initialization, parallel functions, wrap-around, and broadcast processor communications.

The computer has preferably  $N^2$  processing units placed in the form of an  $N$  by  $N$  matrix that has been folded along the diagonal and made up of single processor diagonal units and dual processor general units that are interconnected by a nearest neighbor with wrap-around interconnection structure. In the computer each processing element or PE is a unit of the matrix. Each processor is identified with a reference notation to the original  $N$  by  $N$  matrix prior to folding that supports the transport of  $N$  by  $N$  matrix algorithms to triangular array algorithms.

6

Prior to folding, each PE has four ports, and there are  $N^2$  processing units each possessing North, South, East and West I/O ports for nearest neighbor with wrap-around communications placed in the form of an  $N$  by  $N$  matrix that has been folded along the diagonal and allows the sharing of the North and South I/O ports with the East and West I/O ports.

For our processor with an  $N$  by  $N$  matrix the way of connecting processors is with a process providing a non-conflicting interprocessor communication's mechanism. For example, a mechanism that utilizes a unidirectional communication strategy between processors can be utilized on the Oracle array processor. The non-conflicting interprocessor communication's mechanism can be obtained by requiring all processors utilize a unidirectional and same direction communication strategy.

With our notation each said processing unit is identified by a two subscript notation  $PE_{column,row}$  in reference to the original  $N$  by  $N$  matrix prior to folding. Accordingly the computing apparatus will have  $K(N^2)$  interconnection wires where  $K$  is the number of wires between processors, which for bit-serial interfaces  $K$  can be one ( $K=1$ ). We support single processor diagonal units. The apparatus has single processor diagonal units, identified as  $PE_{i,j}$ , including data storage elements, an execution unit, a broadcast interface for the communications of instructions and data, a data storage interface supporting initialization, and a nearest-neighbor with wrap-around interface, termed the interprocessor interface, and communication's means.

We have also provided new facilities for computation, and these are described in the detail below.

These and other improvements are set forth in the following detailed description. For a better understanding of the invention with advantages and features, reference may be had to the description and to the drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a  $8 \times 8$  mesh for convolution image processing.

FIG. 2 shows a mesh PE internal structure.

FIG. 3 shows an  $8 \times 8$  mesh in accordance with our preferred embodiment.

FIGS. 4A, 4B, 4C, 4D, 4E, 4F, 4G, 4H, 4I, and 4J show a symmetric PE "cell" structure supporting transposition and convolution.

FIGS. 5A and 5B show an  $8 \times 8$  mesh redrawn with our preferred PE symbolism.

FIGS. 6A and 6B show an  $8 \times 8$  matrix  $P$  on square mesh (A) and on our preferred structure (B).

FIGS. 7A and 7B show an  $8 \times 8$  matrix transposed matrix  $P$  on square mesh (A) and on our preferred structure (B).

FIG. 8 shows a convolution window.

FIG. 9 shows a generalized mesh for convolution image processing.

FIG. 10 shows a convolution for pixel P45—Steps 1 through 4.

FIG. 11 shows a convolution for pixel P45—Steps 5 through 8.

FIG. 12 shows a convolution for pixel P45—Step 9.

FIGS. 13A and 13B show a mesh superimposed over a region R.

FIGS. 14A, 14B, 14C, 14D and 14E show a our symmetric PE "cell" structure supporting a finite difference method.

FIG. 15 shows a our finite difference method Steps 1 through 5.

US 6,405,185 B1

7

(Note: For convenience of illustration, FIGURES may be separated in parts and as a convention we place the top of the FIGURE as the first sheet, with subsequent sheets proceeding down and across when viewing the FIGURE, in the event that multiple sheets are used.)

Our detailed description explains the preferred embodiments of our invention, together with advantages and features, by way of example with reference to the above drawings.

## DETAILED DESCRIPTION OF THE INVENTION

### INTRODUCTION

Referring now to the Figures, an example mesh computer is depicted in FIG. 1. To minimize wiring complexity, bit serial interfaces between PEs are assumed. In FIG. 1, the processing elements are labeled as  $PE_{ij}$  where "i" denotes the matrix column and "j" denotes the matrix row. Each  $PE_{ij}$  processing element contains four interface ports labeled North (N), East (E), South (S), and West (W). With wrap-around connections, i.e. a torus configuration, each row contains N interconnection wires and with N rows, there are  $N^2$  horizontal interconnection wires. Each column contains N interconnection wires and with N columns, there are  $N^2$  vertical interconnection wires. The total number of wires in the mesh connected computer with wraparound connections is  $2N^2(K)$ , where K is equal to the number of interprocessor interconnection wires which for bit-serial interfaces K can be equal to 1.

In accordance with our invention we have preferred to describe our invention with bit-serial interfaces. We should here note that it will be possible to have effective parallel interfaces through other expedients. For instance, the application of Howard Olinowich, discloses a way for use of a protocol to interface bit-serial communication with parallel communication in a multi-system environment as described in U.S. Ser. No. 07/799,602, filed Nov. 27, 1991, in his application entitled: "Multi-Media Serial Line Switching Adapter for Parallel Networks and Heterogenous and Homologous Computer Systems". This application is incorporated by reference.

With appropriate network connections our machine may be employed with systems which allow dynamic switching between MIMD, SIMD, and SISD modes, as described in U.S. Ser. No. 07/798,788, filed Nov. 27, 1991, by P. M. Kogge, in his application entitled: "Dynamic Multi-mode Parallel Processor Array Architecture". Our PE can form part of a parallel array processor composed of many nodes, each node having its PE and Memory, and ports to communicate externally with other nodes.

FIG. 2 depicts a typical mesh PE for use in implementing the image processing convolution algorithm as adapted from Lee (Additional Reference 5) with a more explicit depiction of the N, S, E, and W transmitting/receiving ports. The PE consists of four internal registers,  $W_{ab}$ ,  $P_{ij}$ ,  $R_{ij}$ , and  $R'_{ij}$ . The  $P_{ij}$  register holds the  $i^{th}$  pixel value as initialized through an image initialization path, which could be through LSSD (See Additional Reference 6) scan paths.

The  $W_{ab}$  register is loaded with an element of the convolution window through the broadcast mechanism and results of calculations are loaded into the  $R'_{ij}$  register. Results transmitted between PEs are received into the  $R_{ij}$  register. The PEs also contain a bit-serial multiplier, indicated by the "\*" and a bit-serial adder indicated by the "+". For communication purposes, each PE contains four input/

8

output ports whose transmission/reception direction are controlled by "Command Control" logic based upon decoded broadcast commands. The "Command Control" logic allows only one port to be transmitting and one port to be receiving at a time. For example, if port "N" is transmitting then port "S" will be receiving and if port "W" is transmitting then port "E" will be receiving. This follows from the restriction of unidirectional transmission of information between PEs.

In the mesh connected computer organization of FIG. 1 and the PE internal structure of FIG. 2, it can be noted that during any transmission of information in the structure only 50% of the PE connection wires are utilized. It is desirable to achieve the same functionality and performance as the mesh connected computer with half the connecting wires since this amounts to a savings of  $N^2$  wires with corresponding savings in chip real estate. The Oracle computer organization achieves this capability. The first step in the process of creating the Oracle organization of PEs is to fold a mesh connected computer along the  $PE_{ij}$  diagonal elements, i.e. " $i=j$ ". For example, folding the  $8 \times 8$  mesh depicted in FIG. 1 results in the Oracle mesh shown in FIG. 3, where the symmetric PEs,  $PE_{ij}$  and  $PE_{ji}$  are placed together. The symmetric PEs share transmission and reception ports, symbolically indicated in FIG. 3 where the top PE's,  $PE_{ij}$  W, S, N, and E ports are shared with the bottom PE's,  $PE_{ji}$  N, E, W, and S ports respectively. This sharing of I/O ports in the symmetric PEs is shown in greater detail in FIG. 4 Oracle Symmetric PE "Cell" Structure. In FIG. 4A, the internal symmetric PE "cell" consists of a common  $W_{ab}$  register for both internal PEs, two pixel registers labeled  $P_{ij}$  and  $P_{ji}$ , a Selector which allows selection of the matrix P or its transpose  $P^T$ , two result registers  $R_{ij}$  and  $R_{ji}$ , and two receiving registers  $R'_j$  and  $R'_i$ . The PEs also contain a bit-serial multiplier, indicated by the "\*" and a bit-serial adder indicated by the "+". Only four I/O ports are utilized in the symmetric dual PE "cell" as controlled from a single "Command Control" logic that decodes broadcast commands. The "Command Control" logic controls not only the direction of the I/O ports but also the setting of the switches linking the I/O ports to the transmitting and receiving registers. The shared I/O ports are indicated as (top PE direction port label)/(bottom PE direction port label) where  $PE_{ij}$  represents the top PE and  $PE_{ji}$  represents the bottom PE. The choice of top and bottom PE notation is for ease of discussion and representation only.

FIGS. 4B through 4E depict the setting of the internal switches in support of the four transmission modes. In FIG. 4B the "Command Control" has set up the switches and transmitters/receivers for a transmission North (N) with reception from the South (S). Both the top and bottom PEs transmit North while both are receiving information from the South. This is indicated by following the arrows in FIG. 4B and the notation "N/..." for the top  $PE_{ij}$  transmission and ".../N" for the bottom  $PE_{ji}$  transmission. The information received is indicated by the arrows in FIG. 4B and the notation "S/..." for the top  $PE_{ij}$  receiving port and ".../S" for the bottom  $PE_{ji}$  receiving port. Using this notation for the four I/O ports in the symmetric PE "cells", a simplified symbology can be constructed for the symmetric PEs as shown in FIG. 5A where  $PE_{ij}$  is the top PE and  $PE_{ji}$  is the bottom PE. Utilization of this symbology in the Oracle organization results in a simplified diagram FIG. 5B indicating the regularity of the wiring between the cells.

The dual processors internal switch consists of eight connection points A, B, C, D, and W, X, Y, and Z, where: point A is connected to processor  $P_{ij}$ 's register  $R_{ij}$  that receives data from the interprocessor interface,



US 6,405,185 B1

9

point B is connected to processor  $P_{ij}$ 's register  $R'_{ij}$  that supplies data to the interprocessor interface,

point C is connected to processor  $P_{ij}$ 's register  $R'_{ij}$  that supplies data to the interprocessor interface

point D is connected to processor  $P_{ij}$ 's register  $R_{ij}$  that receives data from the interprocessor interface,

point W is connected to receiving/transmitting mechanisms for the transmission and reception of data between the  $PE_{ij}$ 's West and  $PE_{ij}$ 's North neighbor PEs,

point X is connected to receiving/transmitting mechanisms for the transmission and reception of data between the  $PE_{ij}$ 's South and  $PE_{ij}$ 's East neighbor PEs,

point Y is connected to receiving/transmitting mechanisms for the transmission and reception of data between the  $PE_{ij}$ 's North and  $PE_{ij}$ 's West neighbor PEs,

point Z is connected to receiving/transmitting mechanisms for the transmission and reception of data between the  $PE_{ij}$ 's East and  $PE_{ij}$ 's South neighbor PEs.

The dual processor switch provides connection/no connection paths between points A, B, C, D and points W, X, Y, and Z dependent upon the switch state. In the one switch state connection paths between points A and W, B and Z, C and X, and D and Y are provided for Transmit East Receive West. In a second switch state connection paths between points A, B and Y, C and W, and D and Z are provided for Transmit North Receive South. In a third switch state connection paths between points A and Y, B and X, C and Z, and D and W are provided for Transmit South Receive North and in a fourth switch state connection paths between points A and Z, B and W, C and Y, and D and X are provided for transmit West Receive East. It should be noted that the receiving/transmitting mechanisms consists of four bi-directional driver/receiver mechanisms each responsive in one state to drive signals from the PE to an attached receiving PE and responsive in another state to receive signals to the PE from an attached transmitting PE and controls are provided to ensure that of the four bi-directional driver/receiver mechanisms only two are simultaneously transmitting data and two are simultaneously receiving data for the four cases of Transmit East Receive West, Transmit North Receive South, Transmit South Receive North, and Transmit West Receive East.

The diagonal PEs, FIG. 4F, share the West/North ports and the South/East ports requiring only two ports per diagonal PE "cell". FIGS. 4G through 4J depict the setting of the internal switches in support of the four transmission modes. The diagonal processor internal switch mechanism consists of four connection points A, B, X, and Y, where point A is connected to the receiving data register  $R_{ij}$ , point B is connected to register  $R'_{ij}$  that supplies data to the interprocessor interface, point X is connected to receiving/transmitting mechanisms for the transmission and reception of data between the diagonal PE's West and North (W/N) neighbor PEs, and point Y is connected to receiving/transmitting mechanisms for the transmission and reception of data between the diagonal PE's South and East (S/E) neighbor PEs. The diagonal switch provides connection/no connection paths between points A, B and points X, Y dependent upon the switch state. In one switch state a connection path between points A and X and between points B and Y is provided for two transmission/reception cases, namely transmission South, through point Y, reception

10

North, through point X, and transmission East, through point Y, reception West, through point X. In a second switch state a connection path between points A and Y and between points B and X is provided for two transmission/reception cases, namely transmission North, through point X, reception South, through point Y, and transmission West, through point X, reception East, through point Y. It should be noted that the receiving/transmitting mechanisms consists of two bi-directional driver/receiver mechanisms each responsive in one state to drive signals from the PE to an attached receiving PE and responsive in another state to receive signals to the PE from an attached transmitting PE and controls are provided to ensure that the bi-directional driver/receiver mechanisms are not both simultaneously driving or both simultaneously receiving data.

The folding of the square mesh structure along the diagonal places the top edge of the square mesh in line with the West edge and the East edge of the square mesh in line with the South edge. The  $2N$  wraparound connections of the square mesh computer organization which are between the North/South edges and the East/West edges of the square array can be shared in the Oracle mesh organization requiring only  $N$  wraparound connections. Counting the Oracle horizontal interconnection wiring and considering the wrap-around connections as horizontal wires results in:

Oracle Horizontal & Wrap around Wires =

$$(1 + 2 + \dots + N)K = \left(N \frac{(N+1)}{2}\right)K$$

Counting the Oracle vertical interconnection wiring results in:

$$\text{Oracle Vertical Wires} = (1 + 2 + \dots + N - 1)K = \left(N \frac{(N+1)}{2}\right)K$$

The total number of Oracle wires is:

Total Number of Oracle Wires =

$$\left(N \frac{(N+1)}{2} + N \frac{(N+1)}{2}\right)K = N^2(K)$$

Where  $K$  is equal to the number of interprocessor interconnection wires which for bit-serial interfaces  $K$  can be 1. The total number of Oracle wires is demonstrated to be exactly one half the number of wires required by the square mesh organization.

Under the restriction of uni-directional information transfer between PEs, the two computer organizations are next demonstrated to be functionally equivalent. Four cases must be reviewed, namely:

1. Transmit North Receive South
2. Transmit South Receive North
3. Transmit East Receive West
4. Transmit West Receive East

It is shown that the destination points for information transmitted between the PEs is the same in Oracle as it is in the mesh connected organization. For a  $PE_{ij}$

1. Transmit North  $PE_{ij} \rightarrow PE_{i,j-1}$
2. Transmit South  $PE_{ij} \rightarrow PE_{i,j+1}$
3. Transmit East  $PE_{ij} \rightarrow PE_{i+1,j}$
4. Transmit West  $PE_{ij} \rightarrow PE_{i-1,j}$

Where: if  $i-1=0$  or  $j-1=0$  then set  $i=N$  or  $j=N$  and if  $i+1>N$  or  $j+1>N$  then set  $i=1$  or  $j=1$  for the wrap-around connections.

## US 6,405,185 B1

11

In Oracle, we will consider first the symmetric dual PE "cells" where in the top PE "cells"  $i < j$  and in the bottom PE "cells"  $i > j$  since the "i" and "j" values are interchanged. This first consideration accounts for all PEs except the diagonal cells which are considered next. For the symmetric dual PE "cells" and where: if  $i-1=0$  or  $j-1=0$  then set  $i=N$  or  $j=N$  and if  $i+1>N$  or  $j+1>N$  then set  $i=1$  or  $j=1$  for the wrap-around connections:

1. Transmit North Receive South:  
PE<sub>ij</sub> transmits on the N/W wire to PE<sub>i,j-1</sub>  
PE<sub>ij</sub> receives on the S/E wire from PE<sub>i,j+1</sub>  
PE<sub>ij</sub> transmits on the W/N wire to PE<sub>i-1,j</sub>  
PE<sub>ij</sub> receives on the E/S wire from PE<sub>i+1,j</sub>
2. Transmit South Receive North:  
PE<sub>ij</sub> transmits on the S/E wire to PE<sub>i,j+1</sub>  
PE<sub>ij</sub> receives on the N/W wire from PE<sub>i,j-1</sub>  
PE<sub>ij</sub> transmits on the E/S wire to PE<sub>i+1,j</sub>  
PE<sub>ij</sub> receives on the W/N wire from PE<sub>i-1,j</sub>
3. Transmit East Receive West:  
PE<sub>ij</sub> transmits on the E/S wire to PE<sub>i,j+1</sub>  
PE<sub>ij</sub> receives on the W/N wire from PE<sub>i-1,j</sub>  
PE<sub>ij</sub> transmits on the S/E wire to PE<sub>i+1,j</sub>  
PE<sub>ij</sub> receives on the N/W wire from PE<sub>i,j-1</sub>
4. Transmit West Receive East:  
PE<sub>ij</sub> transmits on the W/N wire to PE<sub>i-1,j</sub>  
PE<sub>ij</sub> receives on the E/S wire from PE<sub>i+1,j</sub>  
PE<sub>ij</sub> transmits on the N/W wire to PE<sub>i,j-1</sub>  
PE<sub>ij</sub> receives on the S/E wire from PE<sub>i,j+1</sub>

For the diagonal "cells" where  $i=j$  the following information transfers occur:

1. Transmit North Receive South:  
PE<sub>ij</sub> transmits on the W/N wire to PE<sub>i,j-1</sub>  
PE<sub>ij</sub> receives on the S/E wire from PE<sub>i,j+1</sub>
2. Transmit South Receive North:  
PE<sub>ij</sub> transmits on the S/E wire to PE<sub>i,j+1</sub>  
PE<sub>ij</sub> receives on the W/N wire from PE<sub>i,j-1</sub>
3. Transmit East Receive West:  
PE<sub>ij</sub> transmits on the E/S wire to PE<sub>i,j+1</sub>  
PE<sub>ij</sub> receives on the W/N wire from PE<sub>i-1,j</sub>
4. Transmit West Receive East:  
PE<sub>ij</sub> transmits on the W/N wire to PE<sub>i-1,j</sub>  
PE<sub>ij</sub> receives on the E/S wire from PE<sub>i+1,j</sub>

In all cases no conflicts occur and the correct destination points remain the same as in the square mesh connected organization of PEs.

## MATRIX TRANSPOSITION

In the transposition of matrix "P", the row vectors of the matrix become the column vectors of the transposition matrix "P<sup>T</sup>". An arbitrary element  $p_{ab}$  of matrix "P" becomes element  $p_{ba}$  in the transposition matrix "P<sup>T</sup>". The diagonal elements remain the same. In Oracle a matrix "P" and its transpose can be easily selected since both the element  $p_{ab}$  and its corresponding element  $p_{ba}$  are present in the dual elements. A selector at the output of the "p" registers allows the use of the elements in "P" or in "P<sup>T</sup>" in specified operations. For example, a "P" matrix is shown in FIG. 6A and as loaded on to Oracle is shown in FIG. 6B.

The transpose of matrix P is P<sup>T</sup> and is shown in FIG. 7A and as loaded onto Oracle is shown in FIG. 7B:

FIG. 8 represents a convolution window. FIG. 9 illustrates image processing convolution on a square mesh. For the image processing task considered in this paper a 2-D convolution with a 3x3 convolution window, FIG. 8, will be assumed. The technique employed on the mesh structure is

12

that proposed by S.-Y. Lee and J. K. Aggarwal. The processing element internal structure is logically depicted in FIG. 2, as adapted from Lee with a more explicit depiction of the N, S, E, W Transmitting/Receiving ports.

Assuming an N×N image is already loaded into the N×N mesh P<sub>ij</sub> registers and the R<sub>ij</sub> registers are set to zero, the convolution algorithm can begin with the broadcast of the first window element W<sub>11</sub>. The following steps are followed for the P<sub>ij</sub> convolution path where each step in the path is depicted with its corresponding step number in FIG. 9: (It should be noted that all PEs calculate the same operations on their pixel values and received partial summation values.)

1. Broadcast W<sub>11</sub> and PE<sub>i,j+1</sub> calculates R1=0+W<sub>11</sub>P<sub>i,j+1</sub> and transfers R1 North.
2. Broadcast W<sub>12</sub> and PE<sub>i,j</sub> calculates R2=R1+W<sub>12</sub>P<sub>i,j</sub> and transfers R2 North.
3. Broadcast W<sub>13</sub> and PE<sub>i,j-1</sub> calculates R3=R2+W<sub>13</sub>P<sub>i,j-1</sub> and transfers R3 East.
4. Broadcast W<sub>23</sub> and PE<sub>i,j-1</sub> calculates R4=R3+W<sub>23</sub>P<sub>i,j-1</sub> and transfers R4 East.
5. Broadcast W<sub>33</sub> and PE<sub>i+1,j-1</sub> calculates R5=R4+W<sub>33</sub>P<sub>i+1,j-1</sub> and transfers R5 South.
6. Broadcast W<sub>32</sub> and PE<sub>i,j</sub> calculates R6=R5+W<sub>32</sub>P<sub>i,j</sub> and transfers R6 South.
7. Broadcast W<sub>31</sub> and PE<sub>i-1,j</sub> calculates R7=R6+W<sub>31</sub>P<sub>i-1,j</sub> and transfers R7 West.
8. Broadcast W<sub>21</sub> and PE<sub>i,j+1</sub> calculates R8=R7+W<sub>21</sub>P<sub>i,j+1</sub> and transfers R8 North.
9. Broadcast W<sub>22</sub> and PE<sub>ij</sub> calculates R9=R8+W<sub>22</sub>P<sub>ij</sub> and stop.

At the end of the nine steps each PE<sub>ij</sub> contains:

$$P_{ij} = R9 = W_{11}P_{i,j+1} + W_{12}P_{i,j} + W_{13}P_{i,j-1} + W_{23}P_{i,j-1} + W_{33}P_{i+1,j-1} + W_{32}P_{i,j} + W_{31}P_{i-1,j} + W_{21}P_{i,j+1} + W_{22}P_{ij}$$

In a similar manner, the typical convolution operations on Oracle are depicted in FIGS. 10 through 12 for pixel P<sub>33</sub>. Assuming an N×N image is already loaded into the P<sub>ij</sub> registers and the R<sub>ij</sub> registers are set to zero, the convolution algorithm can begin with the broadcast of the first window element W<sub>11</sub>. The following steps are followed for the P<sub>ij</sub> convolution path where each step in the path is depicted with its corresponding step number in FIGS. 10 through 12: (It should be noted that all PEs calculate the same operations on their pixel values and received partial summation values.)

1. Broadcast W<sub>11</sub> and PE<sub>i,j+1</sub> calculates R1=0+W<sub>11</sub>P<sub>i,j+1</sub> and transfers R1 on the N/W wire.
2. Broadcast W<sub>12</sub> and PE<sub>i,j</sub> calculates R2=R1+W<sub>12</sub>P<sub>i,j</sub> and transfers R2 on the N/W wire.
3. Broadcast W<sub>13</sub> and PE<sub>i,j-1</sub> calculates R3=R2+W<sub>13</sub>P<sub>i,j-1</sub> and transfers R3 on the E/S wire.
4. Broadcast W<sub>23</sub> and PE<sub>i,j-1</sub> calculates R4=R3+W<sub>23</sub>P<sub>i,j-1</sub> and transfers R4 on the E/S wire.
5. Broadcast W<sub>33</sub> and PE<sub>i+1,j-1</sub> calculates R5=R4+W<sub>33</sub>P<sub>i+1,j-1</sub> and transfers R5 on the S/E wire.
6. Broadcast W<sub>32</sub> and PE<sub>i,j</sub> calculates R6=R5+W<sub>32</sub>P<sub>i,j</sub> and transfers R6 on the S/E wire.
7. Broadcast W<sub>31</sub> and PE<sub>i-1,j</sub> calculates R7=R6+W<sub>31</sub>P<sub>i-1,j</sub> and transfers R7 on the W/N wire.
8. Broadcast W<sub>21</sub> and PE<sub>i,j+1</sub> calculates R8=R7+W<sub>21</sub>P<sub>i,j+1</sub> and transfers R8 on the N/W wire.
9. Broadcast W<sub>22</sub> and PE<sub>ij</sub> calculates R9=R8+W<sub>22</sub>P<sub>ij</sub> and stop.

US 6,405,185 B1

13

At the end of the nine steps each  $PE_{ij}$  contains:

$$P_{ij} = R(0) + W_{ij}(P_{i-1,j} + W_{ij}(P_{i-2,j} + W_{ij}(P_{i-3,j} + W_{ij}(P_{i-4,j} + W_{ij}(P_{i-5,j} + W_{ij}(P_{i-6,j} + W_{ij}(P_{i-7,j} + W_{ij}(P_{i-8,j} + W_{ij}(P_{i-9,j}))))))$$

Which for pixel P45 is:

$$G(P45) = W11P36 + W12P35 + W13P34 + W23P44 + W33P54 + W32P55 + W31P56 + W21P46 + W22P45$$

## FINITE DIFFERENCE METHOD EXAMPLE

Finite difference methods for solving differential equations are widely used in a number of scientific and engineering fields such as physics, mechanical engineering, and electrical engineering. In such methods, the derivatives of a differential equation are approximated by difference quotients that may be obtained from a truncated Taylor series (Additional Reference 7).

In the finite difference method consider a second-order partial differential equation, equation (1):

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + D \frac{\partial u}{\partial x} + E \frac{\partial u}{\partial y} + Fu = G \quad (1)$$

Here A, B, C, D, E, F, and G are functions of x and y and continuous in a region R with a boundary S. The function  $u(x,y)$  must be continuous in both R and S.

In finite difference methods, a mesh is superimposed over the region R (as shown in FIG. 13a) and the differential equation (1) is replaced by a difference equation at each mesh point. The partial derivatives are replaced by central difference quotients equations 2a-2d (FIG. 13b).

$$\frac{\partial u}{\partial x} = [u(x+h_x, y) - u(x-h_x, y)] / (2h_x) \quad (2a)$$

$$\frac{\partial u}{\partial y} = [u(x, y+h_y) - u(x, y-h_y)] / (2h_y) \quad (2b)$$

$$\frac{\partial^2 u}{\partial x^2} = [u(x+h_x, y) + u(x-h_x, y) - 2u(x, y)] / h_x^2 \quad (2c)$$

$$\frac{\partial^2 u}{\partial y^2} = [u(x, y+h_y) + u(x, y-h_y) - 2u(x, y)] / h_y^2 \quad (2d)$$

Where  $h_x$  and  $h_y$  are the mesh spacing in the x and y axes respectively, FIG. 13b. Usually, the mesh spacing in both the horizontal and vertical directions is the same, equation 3:

$$h = h_x = h_y \quad (3)$$

Substituting equations (2a) through (2d) in equation (1) with  $R=0$  and multiplying by  $-h^2$ , the following equation is obtained:

$$a_1 u(x,y) - a_2 [u(x+h,y) + u(x-h,y)] - a_3 [u(x,y+h) + u(x,y-h)] - a_4 u(x,y) = 0 \quad (4)$$

Where:

$$a_1 = A(x, y) + \frac{h}{2} D(x, y) \quad (5a)$$

$$a_2 = C(x, y) + \frac{h}{2} E(x, y) \quad (5b)$$

$$a_3 = A(x, y) - \frac{h}{2} D(x, y) \quad (5c)$$

14

-continued

$$a_4 = F(x, y) + \frac{h}{2} G(x, y) \quad (5d)$$

$$a_5 = a_1 + a_2 + a_3 + a_4 - h^2 F(x, y) \quad (5e)$$

$$R(x,y) = -h^2 G(x,y) \quad (5f)$$

10 If Laplace's equation is considered:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (6)$$

Where:  $A=C=1$  and  $B=D=E=F=G=0$ 

A unique solution may be obtained when the boundary conditions S in the region R are given. Laplace's equation may be expressed as small differences. By substitution, the following equation (7a) is obtained:

$$4u(x,y) - u(x+h,y) - u(x-h,y) - u(x,y+h) - u(x,y-h) = 0 \quad (7a)$$

Thus the value of  $u(x,y)$  can be computed by means of an iterative method using the formula (7b):

$$u(x,y) = [u(x+h,y) + u(x-h,y) + u(x,y+h) + u(x,y-h)] / 4 \quad (7b)$$

Laplace's and Poisson's equations are widely used in many theoretical and application problems, such as the analysis of vibrating membranes (Additional Reference 8).

Equation (7b) can be computed on Oracle by mapping equation (7b) to equation 8 using the Oracle notation as presented in the discussion of FIG. 4.

$$P(i,j) = [P(i+1,j) + P(i-1,j) + P(i,j+1) + P(i,j-1)] / 4 \quad (8)$$

This notation is continued for continuity, though the PE internal structure, FIG. 4A, is modified for the finite difference method algorithm, with the modified PE shown in FIG. 14A. The diagonal PEs are also modified from the form shown in FIG. 4F in a similar manner to the dual PE structure. The diagonal PEs contain one execution unit and registers appropriate for a single PE, the structure is inferred from FIG. 14A. The same transmission/receiving I/O ports and controls are utilized in the new PE internal structure, but the computation functions are modified. An adder with an accumulator/shifter is utilized to provide the summation and division by 4 operations required by equation 8. The  $W_{ab}$  register stores a maximum acceptable error value for use by the compare logic. After a new  $P(i,j)$  value is computed it is compared to the previous  $P(i,j)$  value and if the difference is greater than the maximum acceptable error in any PE the calculations must be continued. Since all PEs must have an error value that is less than the maximum acceptable error, the controlling system must know the state of the PEs compare operation. This can be obtained by sending a logic value from each PE through the mesh structure to the controlling system for a global analysis. FIG. 14B through 14E presents the switch and I/O port configurations for the North, South, East, and West transmission modes.

The initial values loaded into the  $P_{ij}$  registers are problem dependent. The accumulators  $R_{ij}$  are initialized to zero and a maximum acceptable error value is loaded into the  $W_{ab}$  register. After initialization the following steps are followed, refer to FIGS. 15 STEPS 1 through 5 corresponding to the following listed steps.

1. Step 1: Transmit North the  $P_{ij}$  values and add the received values to the  $R_{ij}$  value.



US 6,405,185 B1

15

2. Step 2: Transmit East the  $P_{ij}$  values and add the received values to the  $R'_{ij}$  value.
3. Step 3: Transmit South the  $P_{ij}$  values and add the received values to the  $R'_{ij}$  value.
4. Step 4: Transmit West the  $P_{ij}$  values and add the received values to the  $R'_{ij}$  value.
5. Step 5: Shift the accumulated value in  $R'_{ij}$  right 2 positions to accomplish the division by 4 and compare the shifted accumulated value  $R'_{ij}$  with the original value  $P_{ij}$  to verify whether the two values are within the maximum specified error. The result of the compare is transmitted to the edge of the array where a global determination of convergence is tested for.

If global convergence has not been reached then the above process is continued until convergence is reached globally.

While we have described our preferred embodiments of our invention, it will be understood that those skilled in the art, both now and in the future, may make various improvements and enhancements which fall within the scope of the claims which follow. These claims should be construed to maintain the proper protection for the invention first disclosed.

What is claimed is:

1. A processor array comprising:

a plurality of processing elements (PEs) arranged in a multidimensional array, the PEs each comprising a plurality of input/output (I/O) ports, each of the I/O ports for transmitting and receiving data;

the PEs each coupled to four adjacent ones of the plurality of PEs each at one of said plurality of I/O ports, and wherein each PE along an edge of the array is wrap-around coupled to another PE of the plurality of PEs along a nonadjacent edge of the array, said four adjacent ones of the PEs are designated as north, south, east and west PEs;

wherein the PEs are coupled in a folded mesh such that pairs of the PEs share said plurality of input/output ports,

16

a PE designated  $PE_{ij}$  shares the input/output ports with a PE designated  $PE_{ji}$ , where column and row subscripts  $i$  and  $j$ , respectively, are nonequal positive integers,

PEs with  $i < j$  are designated top PEs,

PEs with  $i > j$  are designated bottom PEs,

PEs with  $i = j$  are designated diagonal PEs, and wherein

the diagonal PEs are each coupled to two adjacent ones of the PEs each at one of said plurality of I/O ports, and each diagonal PE at a corner of the array is wraparound coupled to a nondiagonal PE at another corner of the array.

2. The processor array according to claim 1, wherein the PEs each further comprise means for transmitting and receiving data in:

a) transmit east/receive west mode for transmitting data to the east PE over one of said I/O ports while receiving data from the west PE over another of the I/O ports;

b) transmit north/receive south mode for transmitting data to the north PE over one of said I/O ports while receiving data from the south PE over another of the I/O ports;

c) transmit south/receive north mode for transmitting data to the south PE over one of the I/O ports while receiving data from the north PE over another of the I/O ports; and

d) transmit west/receive east mode for transmitting data to the west PE over one of the I/O ports while receiving data from the east PE over another of the I/O ports.

3. The processor array according to claim 2 further comprising means of processing a square matrix algorithms, whereby the array is interconnected using half as many PE interconnections as a two dimensional square array.

4. The processor array according to claim 2, wherein said I/O ports each comprise a bit serial communication scheme.

\* \* \* \* \*

**Electronic Acknowledgement Receipt**

<b>EFS ID:</b>	16678389
<b>Application Number:</b>	13849606
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	6059
<b>Title of Invention:</b>	Processing with Compact Arithmetic Processing Element
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates
<b>Customer Number:</b>	24208
<b>Filer:</b>	Robert Plotkin/Karen Del Greco
<b>Filer Authorized By:</b>	Robert Plotkin
<b>Attorney Docket Number:</b>	A0006-1001C2
<b>Receipt Date:</b>	24-AUG-2013
<b>Filing Date:</b>	25-MAR-2013
<b>Time Stamp:</b>	00:15:52
<b>Application Type:</b>	Utility under 35 USC 111(a)

**Payment information:**

Submitted with Payment	no
------------------------	----

**File Listing:**

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Form (SB08)	rplotkincom-A0006-1001C2_IDS_1449.pdf	64341 203cc7b4801599de3cd5d19a5af07b20d5c7ed0e	no	2

**Warnings:****Information:**



This is not an USPTO supplied IDS fillable form

2	Foreign Reference	rplotkincom-A0006-1001C2-CITE-1_.pdf	16253981	no	60
			59905b465a7d4beab3638b6c7871ab6eb7dd3c65		

**Warnings:****Information:**

<b>Total Files Size (in bytes):</b>	16318322
-------------------------------------	----------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

**New Applications Under 35 U.S.C. 111**

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

**National Stage of an International Application under 35 U.S.C. 371**

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

**New International Application Filed with the USPTO as a Receiving Office**

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

<b>PATENT APPLICATION FEE DETERMINATION RECORD</b>						Application or Docket Number 13/849,606			
Substitute for Form PTO-875									
<b>APPLICATION AS FILED - PART I</b>									
(Column 1)		(Column 2)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA	RATE(\$)	FEE(\$)	RATE(\$)	FEE(\$)			
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	70	N/A				
SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A	N/A	300	N/A				
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	360	N/A				
TOTAL CLAIMS (37 CFR 1.16(j))	42	minus 20 = *	22	x 40 =	880	OR			
INDEPENDENT CLAIMS (37 CFR 1.16(h))	5	minus 3 = *	2	x 210 =	420				
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			0.00					
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				0.00					
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL	2030	TOTAL			
<b>APPLICATION AS AMENDED - PART II</b>									
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY	
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)		
Total (37 CFR 1.16(i))	*	Minus	**	x	=	OR	x	=	
Independent (37 CFR 1.16(h))	*	Minus	***	x	=	OR	x	=	
Application Size Fee (37 CFR 1.16(s))						OR			
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR			
				TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE		
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OR OTHER THAN SMALL ENTITY	
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)		
Total (37 CFR 1.16(i))	*	Minus	**	x	=	OR	x	=	
Independent (37 CFR 1.16(h))	*	Minus	***	x	=	OR	x	=	
Application Size Fee (37 CFR 1.16(s))						OR			
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						OR			
				TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE		
<p>* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.</p> <p>** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".</p> <p>*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".</p> <p>The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.</p>									



## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
 United States Patent and Trademark Office  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
13/849,606	03/25/2013	2193	2100	A0006-1001C2	42	5

CONFIRMATION NO. 6059

## UPDATED FILING RECEIPT

24208  
 ROBERT PLOTKIN, PC  
 15 New England Executive Office Park  
 Burlington, MA 01803



\*OC000000063042988\*

Date Mailed: 08/07/2013

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

## Inventor(s)

Joseph Bates, Newton, MA;

## Applicant(s)

Singular Computing LLC, Newton, MA

**Power of Attorney:** The patent practitioners associated with Customer Number 24208

**Domestic Priority data as claimed by applicant**

This application is a CON of 13/399,884 02/17/2012 PAT 8407273  
 which is a CON of 12/816,201 06/15/2010 PAT 8150902  
 which claims benefit of 61/218,691 06/19/2009

**Foreign Applications** for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.) - None.

*Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.*

Permission to Access - A proper **Authorization to Permit Access to Application by Participating Offices** (PTO/SB/39 or its equivalent) has been received by the USPTO.

**Projected Publication Date:** To Be Determined - pending completion of Security Review

**Non-Publication Request:** No

**Early Publication Request:** No

**\*\* SMALL ENTITY \*\***

**Title**

Processing with Compact Arithmetic Processing Element

**Preliminary Class**

708

**Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No**

**PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES**

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

**LICENSE FOR FOREIGN FILING UNDER**  
**Title 35, United States Code, Section 184**  
**Title 37, Code of Federal Regulations, 5.11 & 5.15**

**GRANTED**

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

**NOT GRANTED**

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

---

***SelectUSA***

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.



ATTORNEY'S DOCKET NO: A0006-1001C2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial No.: 13/849,606  
Confirmation No.: 6059  
Title: Processing with Compact Arithmetic  
Processing Element  
First Inventor: BATES, Joseph  
Filing Date: March 25, 2013  
  
Examiner: N/A  
Art Unit: N/A

**RESPONSE TO NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL**  
**APPLICATION**

In response to the Notice to File Missing Parts of Nonprovisional Application mailed on May 30, 2013, Applicant hereby submits a substitute specification with 1.5 line spacing in compliance with 37 CFR 1.52(b).

No other amendments have been made to the substitute specification, and the substitute specification contains no new matter.

Applicant therefore respectfully requests that the attached substitute specification be deemed to complete the reply to the Notice to File Missing Parts of Nonprovisional Application mailed on May 30, 2013.

Application Serial No. 13/849,606

Attorney Docket No. A0006-1001C2

**CONCLUSIONS**

If the Examiner wishes to discuss this Response, the Examiner is requested to call the Applicant's attorney at the phone number listed below.

No admission is made herein, explicitly or implicitly, that any amendments made herein are made for reasons of patentability.

Please charge any fees or make any credits, to Deposit Account No. 50/1797.

Respectfully submitted,

/Robert Plotkin/

Robert Plotkin, Esq.

Reg. No. 43,861

July 30, 2013

Date

Robert Plotkin, P.C.

15 New England Executive Park

Burlington, MA 01803 USA

Tel: (978) 318-9914

Fax: (978) 318-9060

ATTORNEY'S DOCKET NO: A0006-1001C2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial No.: 13/849,606  
Confirmation No.: 6059  
Title: Processing with Compact Arithmetic  
Processing Element  
First Inventor: BATES, Joseph  
Filing Date: March 25, 2013  
  
Examiner: N/A  
Art Unit: N/A

**RESPONSE TO NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL**  
**APPLICATION**

In response to the Notice to File Missing Parts of Nonprovisional Application mailed on May 30, 2013, Applicant hereby submits a substitute specification with 1.5 line spacing in compliance with 37 CFR 1.52(b).

No other amendments have been made to the substitute specification, and the substitute specification contains no new matter.

Applicant therefore respectfully requests that the attached substitute specification be deemed to complete the reply to the Notice to File Missing Parts of Nonprovisional Application mailed on May 30, 2013.

Application Serial No. 13/849,606

Attorney Docket No. A0006-1001C2

**CONCLUSIONS**

If the Examiner wishes to discuss this Response, the Examiner is requested to call the Applicant's attorney at the phone number listed below.

No admission is made herein, explicitly or implicitly, that any amendments made herein are made for reasons of patentability.

Please charge any fees or make any credits, to Deposit Account No. 50/1797.

Respectfully submitted,

/Robert Plotkin/

Robert Plotkin, Esq.  
Reg. No. 43,861

July 30, 2013

Date

Robert Plotkin, P.C.  
15 New England Executive Park  
Burlington, MA 01803 USA  
Tel: (978) 318-9914  
Fax: (978) 318-9060

Attorney Docket No. A0006-1001C2

**APPLICATION FOR  
UNITED STATES LETTERS PATENT**

**Title:       Processing with Compact Arithmetic  
              Processing Element**

**Inventor(s):   Joseph Bates**



## **Processing with Compact Arithmetic Processing Element**

### **Cross-Reference to Related Applications**

**[0001]** This application is a continuation of U.S. Patent Application Serial Number 13/399,884, filed on February 17, 2012, entitled, “Processing with Compact Arithmetic Processing Element,” now U.S. Patent Number 8,407,273 (Attorney Docket No. A0006-1001C1); which is a continuation of U.S. Patent Application Serial Number 12/816,201, filed on June 15, 2010, entitled, “Processing with Compact Arithmetic Processing Element,” now U.S. Patent Number 8,150,902 (Attorney Docket No. A0006-1001); which claims the benefit of U.S. Provisional Patent Application Serial Number 61/218,691, filed on June 19, 2009, entitled, “Massively Parallel Processing with Compact Arithmetic Element” (Attorney Docket No. A0006-1001L); all of which are hereby incorporated by reference herein.

### **Copyright Notice**

**[0002]** A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

### **Background**

**[0003]** The ability to compute rapidly has become enormously important to humanity. Weather and climate prediction, medical applications (such as drug design and non-invasive imaging), national defense, geological exploration, financial modeling, Internet search, network communications, scientific research in varied fields, and even the design of new computing hardware have each become dependent on the ability to rapidly perform massive amounts of calculation. Future progress, such as the computer-aided design of

Attorney Docket No. A0006-1001C2

complex nano-scale systems or development of consumer products that can see, hear, and understand, will demand economical delivery of even greater computing power.

**[0004]** Gordon Moore's prediction, that computing performance per dollar would double every two years, has proved valid for over 30 years and looks likely to continue in some form. But despite this rapid exponential improvement, the reality is that the inherent computing power available from silicon has grown far more quickly than it has been made available to software. In other words, although the theoretical computing power of computing hardware has grown exponentially, the interfaces through which software is required to access the hardware limits the ability of software to use hardware to perform computations at anything approaching the hardware's theoretical maximum computing power.

**[0005]** Consider a modern silicon microprocessor chip containing about one billion transistors, clocked at roughly 1 GHz. On each cycle the chip delivers approximately one useful arithmetic operation to the software it is running. For instance, a value might be transferred between registers, another value might be incremented, perhaps a multiply is accomplished. This is not terribly different from what chips did 30 years ago, though the clock rates are perhaps a thousand times faster today.

**[0006]** Real computers are built as physical devices, and the underlying physics from which the machines are built often exhibits complex and interesting behavior. For example, a silicon MOSFET transistor is a device capable of performing interesting non-linear operations, such as exponentiation. The junction of two wires can add currents. If configured properly, a billion transistors and wires should be able to perform some significant fraction of a billion interesting computational operations within a few propagation delays of the basic components (a "cycle" if the overall design is a traditional digital design). Yet, today's CPU chips use their billion transistors to enable software to perform merely a few such operations per cycle, not the significant fraction of the billion that might be possible.

### **Summary**

**[0007]** Embodiments of the present invention are directed to a processor or other device, such as a programmable and/or massively parallel processor or other device, which includes processing elements designed to perform arithmetic operations (possibly but not necessarily including, for example, one or more of addition, multiplication, subtraction, and division) on numerical values of low precision but high dynamic range ("LPHDR arithmetic"). Such a processor or other device may, for example, be implemented on a single chip. Whether or not implemented on a single chip, the number of LPHDR arithmetic elements in the processor or other device in certain embodiments of the present invention significantly exceeds (e.g., by at least 20 more than three times) the number of arithmetic elements in the processor or other device which are designed to perform high dynamic range arithmetic of traditional precision (such as 32 bit or 64 bit floating point arithmetic).

**[0008]** In some embodiments, "low precision" processing elements perform arithmetic operations which produce results that frequently differ from exact results by at least .1% (one tenth of one percent). This is far worse precision than the widely used IEEE 754 single precision floating point standard. Programmable embodiments of the present invention may be programmed with algorithms that function adequately despite these unusually large relative errors. In some embodiments, the processing elements have "high dynamic range" in the sense that they are capable of operating on inputs and/or producing outputs spanning a range at least as large as from one millionth to one million.

### **Brief Description of the Drawings**

**[0009]** FIG. 1 is an example overall design of a SIMD processor according to one embodiment of the present invention.

**[0010]** FIG. 2 is an example of the Processing Element Array of a SIMD processor according to one embodiment of the present invention.

**[0011]** FIG. 3 is an example of how a Processing Element in a Processing Element Array communicates data with other parts of the processor

Attorney Docket No. A0006-1001C2

according to one embodiment of the present invention.

**[0012]** FIG. 4 is an example design for a Processing Element according to one embodiment of the present invention.

**[0013]** FIG. 5 is an example LPHDR data word format according to one embodiment of the present invention.

**[0014]** FIG. 6 is an example design for an LPHDR arithmetic unit according to one embodiment of the present invention.

**[0015]** FIG. 7 is an original image.

**[0016]** FIG. 8 is an image blurred by a blur kernel according to one embodiment of the present invention.

**[0017]** FIG. 9 is an image produced by Richardson Lucy deconvolution using floating point arithmetic according to one embodiment of the present invention.

**[0018]** FIG. 10 is an image produced by Richardson Lucy deconvolution using LPHDR floating point arithmetic with added noise (fp+noise) according to one embodiment of the present invention.

**[0019]** FIG. 11 is an image produced by Richardson Lucy deconvolution using LPHDR logarithmic arithmetic (lns) according to one embodiment of the present invention.

### **Detailed Description**

**[0020]** As described above, today's CPU chips make inefficient use of their transistors. For example, a conventional CPU chip containing a billion transistors might enable software to perform merely a few operations per clock cycle. Although this is highly inefficient, those having ordinary skill in the art design CPUs in this way for what are widely accepted to be valid reasons. For example, such designs satisfy the (often essential) requirement for software compatibility with earlier designs. Furthermore, they deliver great precision, performing exact arithmetic with integers typically 32 or 64 bits long and performing rather accurate and widely standardized arithmetic with 32 and 64 bit floating point numbers. Many applications need this kind of precision. As a

result, conventional CPUs typically are designed to provide such precision, using on the order of a million transistors to implement the arithmetic operations.

**[0021]** There are many economically important applications, however, which are not especially sensitive to precision and that would greatly benefit, in the form of application performance per transistor, from the ability to draw upon a far greater fraction of the computing power inherent in those million transistors. Current architectures for general purpose computing fail to deliver this power.

**[0022]** Because of the weaknesses of conventional computers, such as typical microprocessors, other kinds of computers have been developed to attain higher performance. These machines include single instruction stream/multiple data stream (SIMD) designs, multiple instruction stream/multiple data stream (MIMD) designs, reconfigurable architectures such as field programmable gate arrays (FPGAs), and graphics processing unit designs (GPUs) which, when applied to general purpose computing, may be viewed as single instruction stream/multiple thread (SIMT) designs.

**[0023]** SIMD machines follow a sequential program, with each instruction performing operations on a collection of data. They come in two main varieties: vector processors and array processors. Vector processors stream data through a processing element (or small collection of such elements). Each component of the data stream is processed similarly. Vector machines gain speed by eliminating many instruction fetch/decode operations and by pipelining the processor so that the clock speed of the operations is increased.

**[0024]** Array processors distribute data across a grid of processing elements (PEs). Each element has its own memory. Instructions are broadcast to the PEs from a central control unit, sequentially. Each PE performs the broadcast instruction on its local data (often with the option to sit idle that cycle). Array processors gain speed by using silicon efficiently—using just one instruction fetch/decode unit to drive many small simple execution units in parallel.

**[0025]** Array processors have been built using fixed point arithmetic at a wide variety of bit widths, such as 1, 4, 8, and wider, and using floating point



Attorney Docket No. A0006-1001C2

arithmetic. Small bit widths allow the processing elements to be small, which allows more of them to fit in the computer, but many operations must be carried out in sequence to perform conventional arithmetic calculations. Wider widths allow conventional arithmetic operations to be completed in a single cycle. In practice, wider widths are desirable. Machines that were originally designed with small bit widths, such as the Connection Machine-1 and the Goodyear Massively Parallel Processor, which each used 1 bit wide processing elements, evolved toward wider data paths to better support fast arithmetic, producing machines such as the Connection Machine-2 which included 32 bit floating point hardware and the MasPar machines which succeeded the Goodyear machine and provided 4 bit processing elements in the MasPar-1 and 32 bit processing elements in the MasPar-2.

**[0026]** Array processors also have been designed to use analog representations of numbers and analog circuits to perform computations. The SCAMP is such a machine. These machines provide low precision arithmetic, in which each operation might introduce perhaps an error of a few percentage points in its results. They also introduce noise into their computations, so the computations are not repeatable. Further, they represent only a small range of values, corresponding for instance to 8 bit fixed point values rather than providing the large dynamic range of typical 32 or 64 bit floating point representations. Given these limitations, the SCAMP was not intended as a general purpose computer, but instead was designed and used for image processing and for modeling biological early vision processes. Such applications do not require a full range of arithmetic operations in hardware, and the SCAMP, for example, omits general division and multiplication from its design.

**[0027]** While SIMD machines were popular in the 1980s, as price/performance for microprocessors improved designers began building machines from large collections of communicating microprocessors. These MIMD machines are fast and can have price/performance comparable to their component microprocessors, but they exhibit the same inefficiency as those components in that they deliver to their software relatively little computation per

transistor.

**[0028]** Field Programmable Gate Arrays (FPGAs) are integrated circuits containing a large grid of general purpose digital elements with reconfigurable wiring between those elements. The elements originally were single digital gates, such as AND and OR gates, but evolved to larger elements that could, for instance, be programmed to map 6 inputs to 1 output according to any Boolean function. This architecture allows the FPGA to be configured from external sources to perform a wide variety of digital computations, which allows the device to be used as a co-processor to a CPU to accelerate computation. However, arithmetic operations such as multiplication and division on integers, and especially on floating point numbers, require many gates and can absorb a large fraction of an FPGA's general purpose resources. For this reason, modern FPGAs often devote a significant portion of their area to providing dozens or hundreds of multiplier blocks, which can be used instead of general purpose resources for computations requiring multiplication. These multiplier blocks typically perform 18 bit or wider integer multiplies, and use many transistors, as similar multiplier circuits do when they are part of a general purpose CPU.

**[0029]** Existing Field Programmable Analog Arrays (FPAAs) are analogous to FPGAs, but their configurable elements perform analog processing. These devices generally are intended to do signal processing, such as helping model neural circuitry. They are relatively low precision, have relatively low dynamic range, and introduce noise into computation. They have not been designed as, or intended for use as, general purpose computers. For instance, they are not seen by those having ordinary skill in the art as machines that can run the variety of complex algorithms with floating point arithmetic that typically run on high performance digital computers.

**[0030]** Finally, Graphics Processing Units (GPUs) are a variety of parallel processor that evolved to provide high speed graphics capabilities to personal computers. They offer standard floating point computing abilities with very high performance for certain tasks. Their computing model is sometimes based on having thousands of nearly identical threads of computing (SIMT),

Attorney Docket No. A0006-1001C2

which are executed by a collection of SIMD-like internal computing engines, each of which is directed and redirected to perform work for which a slow external DRAM memory has provided data. Like other machines that implement standard floating point arithmetic, they use many transistors for that arithmetic. They are as wasteful of those transistors, in the sense discussed above, as are general purpose CPUs.

**[0031]** Some GPUs include support for 16 bit floating point values (sometimes called the “Half” format). The GPU manufacturers, currently such as NVIDIA or AMD/ATI, describe this capability as being useful for rendering images with higher dynamic range than the usual 32 bit RGBA format, which uses 8 bits of fixed point data per color, while also saving space over using 32 bit floating point for color components. The special effects movie firm Industrial Light and Magic (ILM) independently defined an identical representation in their OpenEXR standard, which they describe as “a high dynamic-range (HDR) image file format developed by Industrial Light & Magic for use in computer imaging applications.” Wikipedia (late 2008) describes the 16 bit floating point representation thusly: “This format is used in several computer graphics environments including OpenEXR, OpenGL, and D3DX. The advantage over 8-bit or 16-bit binary integers is that the increased dynamic range allows for more detail to be preserved in highlights and shadows. The advantage over 32-bit single precision binary formats is that it requires half the storage and bandwidth.”

**[0032]** When a graphics processor includes support for 16 bit floating point, that support is alongside support for 32 bit floating point, and increasingly, 64 bit floating point. That is, the 16 bit floating point format is supported for those applications that want it, but the higher precision formats also are supported because they are believed to be needed for traditional graphics applications and also for so called “general purpose” GPU applications. Thus, existing GPUs devote substantial resources to 32 (and increasingly 64) bit arithmetic and are wasteful of transistors in the sense discussed above.

**[0033]** The variety of architectures mentioned above are all attempts to get more performance from silicon than is available in a traditional processor

design. But designers of traditional processors also have been struggling to use the enormous increase in available transistors to improve performance of their machines. These machines often are required, because of history and economics, to support large existing instruction sets, such as the Intel x86 instruction set. This is difficult, because of the law of diminishing returns, which does not enable twice the performance to be delivered by twice the transistor count. One facet of these designers' struggle has been to increase the precision of arithmetic operations, since transistors are abundant and some applications could be sped up significantly if the processor natively supported long (e.g., 64 bit) numbers. With the increase of native fixed point precision from 8 to 16 to 32 to 64 bits, and of floating point from 32 to 64 and sometimes 128 bits, programmers have come to think in terms of high precision and to develop algorithms based on the assumption that computer processors provide such precision, since it comes as an integral part of each new generation of silicon chips and thus is "free."

**[0034]** Embodiments of the present invention efficiently provide computing power using a fundamentally different approach than those described above. In particular, embodiments of the present invention are directed to computer processors or other devices which use low precision high dynamic range (LPHDR) processing elements to perform computations (such as arithmetic operations).

**[0035]** One variety of LPHDR arithmetic represents values from one millionth up to one million with a precision of about 0.1%. If these values were represented and manipulated using the methods of floating point arithmetic, they would have binary mantissas of no more than 10 bits plus a sign bit and binary exponents of at least 5 bits plus a sign bit. However, the circuits to multiply and divide these floating point values would be relatively large. One example of an alternative embodiment is to use a logarithmic representation of the values. In such an approach, the values require the same number of bits to represent, but multiplication and division are implemented as addition and subtraction, respectively, of the logarithmic representations. Addition and subtraction may be

Attorney Docket No. A0006-1001C2

implemented efficiently as described below. As a result, the area of the arithmetic circuits remains relatively small and a greater number of computing elements can be fit into a given area of silicon. This means the machine can perform a greater number of operations per unit of time or per unit power, which gives it an advantage for those computations able to be expressed in the LPHDR framework.

**[0036]** Another embodiment is to use analog representations and processing mechanisms. Analog implementation of LPHDR arithmetic has the potential to be superior to digital implementation, because it tends to use the natural analog physics of transistors or other physical devices instead of using only the digital subset of the device's behavior. This fuller use of the devices' natural abilities may permit smaller mechanisms for doing LPHDR arithmetic. In recent years, in the field of silicon circuitry, analog methods have been supplanted by digital methods. In part, this is because of the ease of doing digital design compared to analog design. Also in part, it is because of the continued rapid scaling of digital technology ("Moore's Law") compared to analog technology. In particular, at deep submicron dimensions, analog transistors no longer work as they had in prior generations of larger-scale technology. This change of familiar behavior has made analog design still harder in recent years. However, digital transistors are in fact analog transistors used in a digital way, meaning digital circuits are really analog circuits designed to attempt to switch the transistors between completely on and completely off states. As scaling continues, even this use of transistors is starting to come face to face with the realities of analog behavior. Scaling of transistors for digital use is expected either to stall or to require digital designers increasingly to acknowledge and work with analog issues. For these reasons, digital embodiments may no longer be easy, reliable, and scalable, and analog embodiments of LPHDR arithmetic may come to dominate commercial architectures.

**[0037]** Because LPHDR processing elements are relatively small, a single processor or other device may include a very large number of LPHDR processing elements, adapted to operate in parallel with each other, and



therefore may constitute a massively parallel LPHDR processor or other device. Such a processor or other device has not been described or practiced as a means of doing general purpose computing by those having ordinary skill in the art for at least two reasons. First, it is commonly believed by those having ordinary skill in the art, that LPHDR computation, and in particular massive amounts of LPHDR computation, whether performed in a massively parallel way or not, is not practical as a substrate for moderately general computing. Second, it is commonly believed by those having ordinary skill in the art that massive amounts of even high precision computation on a single chip or in a single machine, as is enabled by a compact arithmetic processing unit, is not useful without a corresponding increase in bandwidth between processing elements within the machine and into and out of the machine because computing is wire limited and arithmetic can be considered to be available at no cost.

**[0038]** Despite these views—that massive amounts of arithmetic on a chip or in a massively parallel machine are not useful, and that massive amounts of LPHDR arithmetic are even worse—embodiments of the present invention disclosed herein demonstrate that massively parallel LPHDR designs are in fact useful and provide significant practical benefits in at least several significant applications.

**[0039]** To conclude, modern digital computing systems provide high precision arithmetic, but that precision is costly. A modern double precision floating point multiplier may require on the order of a million transistors, even though only a handful of transistors is required to perform a low precision multiplication. Despite the common belief among those having ordinary skill in the art that modern applications require high precision processing, in fact a variety of useful algorithms function adequately at much lower precision. As a result, such algorithms may be performed by processors or other devices implemented according to embodiments of the present invention, which come closer to achieving the goal of using a few transistors to multiply and a wire junction to add, thus enabling massively parallel arithmetic computation to be performed with relatively small amounts of physical resources (such as a single

silicon chip). Although certain specialized tasks can function at low precision, it is not obvious, and in fact has been viewed as clearly false by those having ordinary skill in the art, that relatively general purpose computing such as is typically performed today on general purpose computers can be done at low precision. However, in fact a variety of useful and important algorithms can be made to function adequately at much lower than 32 bit precision in a massively parallel computing framework, and certain embodiments of the present invention support such algorithms, thereby offering much more efficient use of transistors, and thereby provide improved speed, power, and/or cost, compared to conventional computers.

**[0040]** Various computing devices implemented according to embodiments of the present invention will now be described. Some of these embodiments may be an instance of a SIMD computer architecture. Other architectures may be used, such as MIMD architectures, programmable array architectures (such as FPGAs and FPAAs), or GPU/SIMT architectures. The techniques disclosed herein may, for example, be implemented using any processor or other device having such an existing architecture, and replacing or augmenting some or all existing arithmetic units in the processor or other device, if any, with LPHDR arithmetic units in any of the ways disclosed herein. Devices implemented according to embodiments of the present invention, however, need not start with an existing processor design, but instead may be designed from scratch to include LPHDR arithmetic units within any of the architectures just described, or any other architecture.

**[0041]** Embodiments of the present invention may, for example, be implemented using the architecture of a particular kind of SIMD computer, the array processor. There are many variations and specific instances of array processors described in the scientific and commercial literature. Examples include the Illiac 4, the Connection Machine 1 and 2, the Goodyear MPP, and the MasPar line of computers.

**[0042]** Embodiments of the present invention need not, however, be implemented as SIMD computers. For example, embodiments of the present

Attorney Docket No. A0006-1001C2

invention may be implemented as FPGAs, FPAAAs, or related architectures that provide for flexible connectivity of a set of processing elements. For example, embodiments of the present invention may be implemented as GPU/SIMTs and as MIMDs, among others. For example, embodiments of the present invention may be implemented as any kind of machine which uses LPHDR arithmetic processing elements to provide computing using a small amount of resources (e.g., transistors or volume) compared with traditional architectures.

Furthermore, references herein to "processing elements" within embodiments of the present invention should be understood more generally as any kind of execution unit, whether for performing LPHDR operations or otherwise.

**[0043]** An example SIMD computing system 100 is illustrated in FIG. 1. The computing system 100 includes a collection of many processing elements (PEs) 104. Sometimes present are a control unit (CU) 106, an I/O unit (IOU) 108, various Peripheral devices 110, and a Host computer 102. The collection of PEs is referred to herein as "the Processing Element Array" (PEA), even though it need not be two-dimensional or an array or grid or other particular layout. Some machines include additional components, such as an additional memory system called the "Staging Memory" in the Goodyear MPP, but these additional elements are neither essential in the computer nor needed to understand embodiments of the present invention and therefore are omitted here for clarity of explanation. One embodiment of the present invention is a SIMD computing system of the kind shown in FIG. 1, in which one or more (e.g., all) of the PEs in the PEA 104 are LPHDR elements, as that term is used herein.

**[0044]** The Host 102 is responsible for overall control of the computing system 100. It performs the serial, or mostly serial, computation typical of a traditional uni-processor. The Host 102 could have more complicated structure, of course, including parallelism of various sorts. Indeed a heterogeneous computing system combining multiple computing architectures in a single machine is a good use for embodiments of the present invention.

**[0045]** A goal of the Host 102 is to have the PEA 104 perform massive amounts of computation in a useful way. It does this by causing the PEs to

Attorney Docket No. A0006-1001C2

perform computations, typically on data stored locally in each PE, in parallel with one another. If there are many PEs, much work gets done during each unit of time.

**[0046]** The PEs in the PEA 104 may be able to perform their individual computations roughly as fast as the Host 102 performs its computations. This means it may be inefficient to have the Host 102 attempt to control the PEA 104 on a time scale as fine as the Host's or PEA's minimal time step. (This minimal time, in a traditional digital design, would be the clock period.) For this reason, the specialized control unit (CU) 106 may be included in the architecture. The CU 106 has the primary task of retrieving and decoding instructions from an instruction memory, which conceptually is part of the CU 106, and issuing the partially decoded instructions to all the PEs in the PEA 104. (This may be viewed by the CU software as happening roughly simultaneously for all the PEs, though it need not literally be synchronous, and in fact it may be effective to use an asynchronous design in which multiple instructions at different stages of completion simultaneously propagate gradually across the PEA, for instance as a series of wave fronts.)

**[0047]** In a design which includes the CU 106, the Host 102 typically will load the instructions (the program) for the PEA 104 into the CU instruction memory (not shown in FIG. 1), then instruct the CU 106 to interpret the program and cause the PEA 104 to compute according to the instructions. The program may, for example, look generally similar to a typical machine language program, with instructions to cause data movement, logical operations, arithmetic operations, etc., in and between the PEs and other instructions to do similar operations together with control flow operations within the CU 106. Thus, the CU 106 may run a typical sort of program, but with the ability to issue massively parallel instructions to the PEA 104.

**[0048]** In order to get data into and out of the CU 106 and PEA 104, the I/O Unit 108 may interface the CU 106 and PEA 104 with the Host 102, the Host's memory (not shown in FIG. 1), and the system's Peripherals 110, such as external storage (e.g., disk drives), display devices for visualization of the

Attorney Docket No. A0006-1001C2

computational results, and sometimes special high bandwidth input devices (e.g., vision sensors). The PEA's ability to process data far faster than the Host 102 makes it useful for the IOU 108 to be able to completely bypass the Host 102 for some of its data transfers. Also, the Host 102 may have its own ways of communicating with the Peripherals 110.

**[0049]** The particular embodiment illustrated in FIG. 1 is shown merely for purposes of example and does not constitute a limitation of the present invention. For example, alternatively the functions performed by the CU 106 could instead be performed by the Host 102 with the CU 106 omitted. The CU 106 could be implemented as hardware distant from the PEA 104 (e.g., off-chip), or the CU 106 could be near to the PEA 104 (e.g., on-chip). I/O could be routed through the CU 106 with the IOU 108 omitted or through the separate I/O controller 108, as shown. Furthermore, the Host 102 is optional; the CU 106 may include, for example, a CPU, or otherwise include components sufficient to replace the functions performed by the Host 102. The Peripherals 110 shown in FIG. 1 are optional. The design shown in FIG. 1 could have a special memory, such as the Goodyear MPP's "staging memory," which provides an intermediate level of local storage. Such memory could, for example, be bonded to the LPHDR chip using 3D fabrication technology to provide relatively fast parallel access to the memory from the PEs in the PEA 104.

**[0050]** The PEA 104 itself, besides communicating with the CU 106 and IOU 108 and possibly other mechanisms, has ways for data to move within the array. For example, the PEA 104 may be implemented such that data may move from PEs only to their nearest neighbors, that is, there are no long distance transfers. FIGS. 2 and 3 show embodiments of the present invention which use this approach, where the nearest neighbors are the four adjacent PEs toward the North, East, West, and South, called a NEWS design. For example, FIG. 2 shows a subset of the PEs in PEA 104, namely PE 202, PE 204, PE 206, PE 208, and PE 210. When the CU 106 issues data movement instructions, all the PEs access data from or send data to their respective specified nearest neighbor. For instance, every PE might access a specified data value in its



neighbor to the West and copy it into its own local storage. In some embodiments, such as some analog embodiments, these kinds of transfers may result in some degradation of the value copied.

**[0051]** FIG. 3 shows a PE 302 that includes data connections to the IOU 108. PE 302 is connected at the North to PE 304, at the East to PE 306, at the South to PE 308, and at the West to PE 310. However, driving signals from inside the PEA 104 out to the IOU 108 usually requires a physically relatively large driving circuit or analogous mechanism. Having those at every PE may absorb much of the available resources of the hardware implementation technology (such as VLSI area). In addition, having independent connections from every PE to the IOU 108 means many such connections, and long connections, which also may absorb much of the available hardware resources. For these reasons, the connections between the PEs and the IOU 108 may be limited to those PEs at the edges of the PE array 104. In this case, to get data out of, and perhaps into, the PEA 104, the data is read and written at the edges of the array and CU instructions are performed to shift data between the edges and interior of the PEA 104. The design may permit data to be pushed from the IOU 108 inward to any PE in the array using direct connections, but may require readout to occur by using the CU 106 to shift data to the edges where it can be read by the IOU 108.

**[0052]** Connections between the CU 106 and PEA 104 have analogous variations. One design may include the ability to drive instructions into all the PEs roughly simultaneously, but another approach is to have the instructions flow gradually (for instance, shift in discrete time steps) across the PEA 104 to reach the PEs. Some SIMD designs, which may be implemented in embodiments of the present invention, have a facility by which a "wired-or" or "wired-and" of the state of every PE in the PEA 104 can be read by the CU 106 in approximately one instruction delay time.

**[0053]** There are many well studied variations on these matters in the literature, any of which may be incorporated into embodiments of the present invention. For example, an interconnect, such as an 8-way local interconnect,

may be used. The local connections may include a mixture of various distance hops, such as distance 4 or 16 as well as distance 1. The outside edges may be connected using any topology, such as a torus or twisted torus. Instead of or in addition to a local interconnect, a more complex global interconnect, such as the hypercube design, may be used. Furthermore, the physical implementation of the PEA 104 (e.g., a chip) could be replicated (e.g., tiled on a circuit board) to produce a larger PEA. The replication may form a simple grid or other arrangement, just as the component PEAs may but need not be grids.

**[0054]** FIG. 4 shows an example design for a PE 400 (which may be used to implement any one or more of the PEs in the PEA 104). The PE 400 stores local data. The amount of memory for the local data varies significantly from design to design. It may depend on the implementation technologies available for fabricating the PE 400. Sometimes rarely changing values (Constants) take less room than frequently changing values (Registers), and a design may provide more Constants than Registers. For instance, this may be the case with digital embodiments that use single transistor cells for the Constants (e.g., floating gate Flash memory cells) and multiple transistor cells for the Registers (e.g., 6-transistor SRAM cells). Sometimes the situation is reversed, as may be the case in analog embodiments, where substantial area for capacitance may be needed to ensure stable long term storage of Constants, and such embodiments may have more Registers than Constants. Typical storage capacities might be tens or hundreds of arithmetic values stored in the Registers and Constants in each PE, but these capacities are adjustable by the designer. Some designs, for instance, may have Register storage but no Constant storage. Some designs may have thousands or even many more values stored in each PE. All of these variations may be reflected in embodiments of the present invention.

**[0055]** Each PE needs to operate on its local data. For this reason within the PE 400 there are data paths 402a-i, routing mechanisms (such as the multiplexor MUX 404), and components to perform some collection of logical and arithmetic operations (such as the logic unit 406 and the LPHDR arithmetic unit

408). The LPHDR arithmetic unit 408 performs LPHDR arithmetic operations, as that term is used herein. The input, output, and intermediate “values” received by, output by, and operated on by the PE 400 may, for example, take the form of electrical signals representing numerical values.

**[0056]** The PE 400 also may have one or more flag bits, shown as Mask 410 in FIG. 4. The purpose of the Mask 410 is to enable some PEs, the ones in which a specified Mask bit is set, to ignore some instructions issued by the CU 106. This allows some variation in the usual lock-step behaviors of all PEs in the PEA 104. For instance, the CU 106 may issue an instruction that causes each PE to reset or set its Mask 410 depending on whether a specified Register in the PE is positive or negative. A subsequent instruction, for instance an arithmetic instruction, may include a bit meaning that the instruction should be performed only by those PEs whose Mask 410 is reset. This combination has the effect of conditionally performing the arithmetic instruction in each PE depending on whether the specified Register in that PE was positive. As with the Compare instructions of traditional computers, there are many possible design choices for mechanisms to set and clear Masks.

**[0057]** The operation of the PEs is controlled by control signals 412a-d received from the CU 106, four of which are shown in FIG. 4 merely for purposes of example and not limitation. We have not shown details of this mechanism, but the control signals 412a-d specify which Register or Constant memory values in the PE 400 or one of its neighbors to send to the data paths, which operations should be performed by the Logic 406 or Arithmetic 408 or other processing mechanisms, where the results should be stored in the Registers, how to set, reset, and use the Mask 410, and so on. These matters are well described in the literature on SIMD processors.

**[0058]** Many variations of this PE 400 and PEA design are possible and fall within the scope of the present invention. Digital PEs can have shifters, lookup tables, and many other mechanisms such as described in the literature. Analog PEs can have time-based operators, filters, comparators with global broadcast signals and many other mechanisms such as described in the

Attorney Docket No. A0006-1001C2

literature. The PEA 104 can include global mechanisms such as wired-OR or wired-AND for digital PEAs or wired-SUM for analog PEAs. Again, there are many variations well described in the literature on digital and analog computing architectures.

**[0059]** For example, LPHDR operations other than and/or in addition to addition and multiplication may be supported. For example, a machine which can only perform multiplication and the function  $(1-X)$  may be used to approximate addition and other arithmetic operations. Other collections of LPHDR operations may be used to approximate LPHDR arithmetic operations, such as addition, multiplication, subtraction, and division, using techniques that are well-known to those having ordinary skill in the art.

**[0060]** One aspect of embodiments of the present invention that is unique is the inclusion of LPHDR arithmetic mechanisms in the PEs. Embodiments of such mechanisms will now be described.

**[0061]** One digital embodiment of the LPHDR arithmetic unit 408 operates on digital (binary) representations of numbers. In one digital embodiment these numbers are represented by their logarithms. Such a representation is called a Logarithmic Number System (LNS), which is well-understood by those having ordinary skill in the art.

**[0062]** In an LNS, numbers are represented as a sign and an exponent. There is an implicit base for the logarithms, typically 2 when working with digital hardware. In the present embodiment, a base of 2 is used for purposes of example. As a result, a value, say  $B$ , is represented by its sign and a base 2 logarithm, say  $b$ , of its absolute value. For numbers to have representation errors of at most, say, 1% (one percent), the fractional part of this logarithm should be represented with enough precision that the least possible change in the fraction corresponds to about a 1% change in the value  $B$ . If fractions are represented using 6 bits, increasing or decreasing the fraction by 1 corresponds to multiplying or dividing  $B$  by the 64th root of 2, which is approximately 1.011. This means that numbers may be represented in the present embodiment with a multiplicative error of approximately 1%. So, in this

example embodiment the fraction part of the representation has 6 bits.

**[0063]** Furthermore, the space of values processed in the present embodiment have high dynamic range. To represent numbers whose absolute value is from, say, one billionth to one billion, the integer part of the logarithm must be long enough to represent plus or minus the base 2 logarithm of one billion. That logarithm is about 29.9. In the present embodiment the integer part of the logarithm representation is 5 bits long to represent values from 0 to 31, which is sufficient. There also is a sign bit in the exponent. Negative logarithms are represented using two's complement representation.

**[0064]** In an LNS, the value zero corresponds to the logarithm negative infinity. One can choose a representation to explicitly represent this special value. However, to minimize resources (for instance, area) used by arithmetic circuits, the present embodiment represents zero by the most negative possible logarithm, which is -32, corresponding to the two's complement bit representation '100000 000000', and denoting a value of approximately  $2.33E-10$ .

**[0065]** When computing, situations can arise in which operations cannot produce reasonable values. An example is when a number is too large to be represented in the chosen word format, such as when multiplying or adding two large numbers or upon divide by zero (or nearly zero). One common approach to this problem is to allow a value to be marked as Not A Number (NAN) and to make sure that each operation produces NAN if a problem arises or if either of its inputs is NAN. The present embodiment uses this approach, as will be described in the following.

**[0066]** FIG. 5 shows the word format 500 for these numbers, in the present embodiment. It has one NAN bit 502a, one bit 502b for the sign of the value, and 12 bits 502c-e representing the logarithm. The logarithm bits include a 5 bit integer part 502d and a 6 bit fraction part 502e. To permit the logarithms to be negative, there is a sign bit 502c for the logarithm which is represented in two's complement form. The NAN bit is set if some problem has arisen in computing the value. The word format 500 shown in FIG. 5 is merely an example and does not constitute a limitation of the present invention. Other



variations may be used, so long as they have low precision and high dynamic range.

**[0067]** FIG. 6 shows an example digital implementation of the LPHDR arithmetic unit 408 for the representation illustrated in FIG. 5. The unit 408 receives two inputs, A 602a and B 602b, and produces an output 602c. The inputs 602a-b and output 602c may, for example, take the form of electrical signals representing numerical values according to the representation illustrated in FIG. 5, as is also true of signals transmitted within the unit 408 by components of the unit 408. The inputs 602a-b and output 602c each are composed of a Value and a NAN (Not A Number) bit. The unit 408 is controlled by control signals 412a-d, coming from the CU 106, that determine which available arithmetic operation will be performed on the inputs 602a-b. In this embodiment, all the available arithmetic operations are performed in parallel on the inputs 602a-b by adder/subtractor 604, multiplier 606, and divider 608. Adder/subtractor 604 performs LPHDR addition and subtraction, multiplier 606 performs LPHDR multiplication, and divider 608 performs LPHDR division.

**[0068]** The desired result (from among the outputs of adder/subtractor 604, multiplier 606, and divider 608) is chosen by the multiplexers (MUXes) 610a and 610b. The right hand MUX 610b sends the desired value to the output 602c. The left hand MUX 610a sends the corresponding NAN bit from the desired operation to the OR gate 612, which outputs a set NAN bit if either input is NAN or if the specified arithmetic operation yields NAN. The computing architecture literature discusses many variations which may be incorporated into the embodiment illustrated in FIG. 6.

**[0069]** LNS arithmetic has the great advantage that multiplication (MUL) and division (DIV) are very easy to compute and take few physical resources (e.g., little area in a silicon implementation). The sign of the result is the exclusive-or of the signs of the operands. The logarithm part of the output is the sum, in the case of MUL, or the difference, in the case of DIV, of the logarithm parts of the operands. The sum or difference of the logarithms can overflow, producing a NAN result. Certain other operations also are easy in LNS

Attorney Docket No. A0006-1001C2

arithmetic. For instance, square root corresponds to dividing the logarithm in half, which in our representation means simply shifting it one bit position to the right.

**[0070]** Thus, the multiplier 606 and divider 608 in FIG. 6 are implemented as circuits that simply add or subtract their inputs, which are two's complement binary numbers (which in turn happen to be logarithms). If there is overflow, they output a 1 for NAN.

**[0071]** Implementing addition and subtraction in LNS, that is, the adder/subtractor 604 in FIG. 6, follows a common approach used in the literature on LNS. Consider addition. If we have two positive numbers B and C represented by their logarithms b and c, the representation of the sum of B and C is  $\log(B+C)$ . An approach to computing this result that is well known to those skilled in the art is based on noticing that  $\log(B+C) = \log(B*(1+C/B)) = \log(B) + \log(1+C/B) = b + F(c-b)$  where  $F(x) = \log(1+2^x)$ . Thus, the present embodiment computes c-b, feeds that through F, and adds the result to b, using standard digital techniques known to those skilled in the art.

**[0072]** Much of the published literature about LNS is concerned with how to compute F(x), the special function for ADD, along with a similar function for SUB. Often these two functions share circuitry, and this is why a single combined adder/subtractor 604 is used in the embodiment of FIG. 6. There are many published ways to compute these functions or approximations to them, including discussions of how to do this when the values are of low precision. Any such method, or other method, may be used. Generally speaking, the more appropriate variations for massively parallel LPHDR arithmetic are those that require the minimal use of resources, such as circuit area, taking advantage of the fact that the representation used in the embodiment of FIG. 6 is low precision and that the arithmetic operations need not be deterministic nor return the most accurate possible answer within the low precision representation. Thus, embodiments of the present invention may use circuitry that does not compute the best possible answer, even among the limited choices available in a low precision representation.

**[0073]** In order to enable conditional operation of selected PEs, the present embodiment is able to reset and set the MASK flag 410 based on results of computations. The mechanism for doing this is that the CU 106 includes instructions that cause the MASK 410 in each PE to unconditionally reset or set its flag along with other instructions to perform basic tests on values entering the MASK 410 on data path 402f and to set the flag accordingly. Examples of these latter instructions include copying the sign bit or NAN bit of the word on data path 402f into the MASK bit 410. Another example is to set the MASK bit 410 if the 12 bit value part of the word on data path 402f is equal to binary zero. There are many additional and alternative ways for doing this that are directly analogous to comparison instructions in traditional processors and which are well understood by those skilled in the art.

**[0074]** It is worth noting that while the obvious method of using the above LNS operations is to do LPHDR arithmetic, the programmer also may consider selected values to be 12 bit two's complement binary numbers. MUL and DIV may be used to add and subtract such values, since that is precisely their behavior in LNS implementations. The Mask setting instructions can compare these simple binary values. So besides doing LPHDR computations, this digital embodiment using LNS can perform simple binary arithmetic on short signed integers.

**[0075]** Some embodiments of the present invention may include analog representations and processing methods. Such embodiments may, for example, represent LPHDR values as charges, currents, voltages, frequencies, pulse widths, pulse densities, various forms of spikes, or in other forms not characteristic of traditional digital implementations. There are many such representations discussed in the literature, along with mechanisms for processing values so represented. Such methods, often called Analog methods, can be used to perform LPHDR arithmetic in the broad range of architectures we have discussed, of which SIMD is one example.

**[0076]** An example of an analog SIMD architecture is the SCAMP design (and related designs) of Dudek. In that design values have low dynamic

range, being accurate roughly to within 1%. Values are represented by charges on capacitors. Those capacitors typically are the gates of transistors. Each PE has several memory cells, analogous to the Registers shown in FIG. 4. Addition is performed by turning on pass transistors from the two operands, which transfer their charge onto an analog bus, where it is summed by the natural physics of charge and wires, upon which it is gated onto another Register to charge up its capacitor, which then represents the sum of the operands. The detailed mechanism disclosed by Dudek actually produces the negative of the sum, but the basic concept is as described and is a simple way to perform addition and subtraction using analog representations and simple processing mechanisms.

**[0077]** Variations of the SCAMP design have been fabricated and used to perform a range of low precision, low dynamic range computations related to image processing. These designs do not perform high dynamic range arithmetic, nor do they include mechanisms for performing multiplication or division of values stored in Registers. However, the Dudek designs suggest the general feasibility of constructing analog SIMD machines. The following describes how to build an analog SIMD machine that performs LPHDR arithmetic, and is thus an embodiment of the present invention.

**[0078]** One embodiment of the present invention represents values as a mixture of analog and digital forms. This embodiment represents values as low precision, normalized, base 2 floating point numbers, where the mantissa is an analog value and the exponent is a binary digital value. The analog value may be accurate to about 1%, following the approach of Dudek, which is well within the range of reasonable analog processing techniques. The exponent may be 6 bits long, or whatever is needed to provide the desired high dynamic range.

**[0079]** To multiply values, the embodiment proceeds by analogy to traditional floating point methods. The digital exponents are summed using a binary arithmetic adder, a standard digital technique. The analog mantissas are multiplied. Since they represent normalized values between approximately  $1/2$  and 1, their product may be as small as approximately  $1/4$ . Such a product value needs to be normalized back to the range  $1/2$  to 1. This is done, in the present

Attorney Docket No. A0006-1001C2

embodiment, by comparing the analog mantissa to an analog representation of  $1/2$ , using a threshold circuit. If the mantissa is below  $1/2$ , then it is doubled and one is subtracted from the exponent, where such subtraction is simple digital subtraction. Doubling the mantissa is implemented in a way that corresponds to the chosen analog representation. For example, whatever means are being used to add two analog values can be used to double the mantissa, by adding it to a copy of itself. For example, if the mantissa is represented as a current, such as copy may be produced by a current mirror, or other suitable mechanism, and addition may be performed by a current summing junction.

**[0080]** The means of multiplying the original analog mantissas depends on the representation chosen. For example, if mantissas are represented using charge, following SCAMP, then any known method from the literature may be used to convert charge to current. For instance, since the charge on a capacitor determines the voltage on the capacitor, this may be implemented as a conversion from voltage to current, which is a basic technique in analog electronics known to those skilled in the art. In any case, if the mantissas are represented as currents, or once the mantissas are converted to currents, they may be multiplied using, for instance, the techniques of Gilbert. The Gilbert multiplier produces a current, representing the product, which may, if necessary, then be converted back to charge (or whatever representation is used). These are merely examples of how the needed operations might be performed. The literature discusses these matters extensively and these kinds of analog circuits are known to those skilled in the art.

**[0081]** Adding and subtracting values requires pre-normalization of the values to the same exponent, as is done in traditional digital floating point arithmetic. The present embodiment does this by comparing the exponents and choosing the smaller one. Then the smaller one is subtracted from the larger, using digital means. The difference specifies how many times the mantissa which corresponds to the smaller exponent needs to be divided in half. If that mantissa is represented by (or converted to) a current, then an analog R-2R style ladder may be used to divide the current in half the required number of times,



Attorney Docket No. A0006-1001C2

with the stage of the ladder specified by the difference of exponents calculated as above. The resulting scaled down current is added to (or subtracted from, if this is an LPHDR subtraction operation) the current corresponding to the mantissa associated with the larger exponent to yield the output mantissa. The output exponent associated with the output mantissa is the larger exponent. Post-normalization may be needed at this point. If the output mantissa is greater than 1, then it needs to be divided in half and the output exponent needs to be incremented. If it is less than  $1/2$ , then it needs to be doubled enough times to exceed  $1/2$  and the output exponent must be decremented correspondingly, which may be performed by a series of threshold circuits, doubler circuits, and associated decrementer circuits. These increments and decrements of the binary digital exponent, and corresponding doublings and halvings of the analog mantissa current, are straightforward operations well known to those skilled in the art.

**[0082]** The present embodiment represents the exponent as a digital binary number. Alternate embodiments may represent the exponent as an analog value. However, it is important that the exponent be represented in storage and computation in such a manner that neither noise nor other errors cause a change in the value it represents. Such changes in the exponent could introduce factors of two (or in some embodiments larger) changes in the values of the stored numbers. To maintain accuracy of the exponents, an embodiment may quantize the exponent to relatively few levels, for instance 16 values plus a sign bit. During processing, slight variations in the analog representation of the exponent may then be removed by circuitry that restores values to the 16 standard quantization levels. To get sufficient dynamic range in such an embodiment, the floating point numbers may be processed as base 4 numbers, rather than the usual base 2 numbers. This means, for instance, that normalized mantissas are in the range  $1/4$  to 1. The methods discussed above for addition, subtraction, and multiplication apply as described, with slight and straightforward variations.

**[0083]** The analog and mixed signal embodiments discussed above

are merely examples and do not constitute a limitation of the present invention. The published literature on neuromorphic, analog, and mixed signal techniques provides a wealth of methods that enable LPHDR storage and processing to be implemented. Such storage and processing may introduce noise as well as fabrication errors into the behavior of machines performing LPHDR arithmetic. The results we present below, on software applications running using “fp+noise” arithmetic, show that despite these very “un-digital” qualities a machine built in this way is surprisingly useful.

**[0084]** Evidence that LPHDR arithmetic is useful in several important practical computing applications will now be provided. The evidence is presented for a broad variety of embodiments of the present invention, thereby showing that the usefulness does not depend much on the detailed implementation.

**[0085]** For the goal of showing usefulness, we choose a very general embodiment of an LPHDR machine. Our model of the machine is that it provides at least the following capabilities: (1) is massively parallel, (2) provides LPHDR arithmetic possibly with noise, (3) provides a small amount of memory local to each arithmetic unit, (4) provides the arithmetic/memory units in a two-dimensional physical layout with only local connections between units (rather than some more powerful, flexible, or sophisticated connection mechanism), and (5) provides only limited bandwidth between the machine and the host machine. Note that this model is merely an example which is used for the purpose of demonstrating the utility of various embodiments of the present invention, and does not constitute a limitation of the present invention. This model includes, among others, implementations that are digital or analog or mixed, have zero or more noise, have architectures which are FPGA-like, or SIMD-like, or MIMD-like, or otherwise meet the assumptions of the model. More general architectures, such as shared memory designs, GPU-like designs, or other sophisticated designs subsume this model's capabilities, and so LPHDR arithmetic in those architectures also is useful. While we are thus showing that LPHDR arithmetic is useful for a broad range of designs, of which SIMD is only an instance, for

purpose of discussion below, we call each unit, which pairs memory with arithmetic, a Processing Element or "PE".

**[0086]** Several applications are discussed below. For each, the discussion shows (1) that the results are useful when computation is performed in possibly noisy LPHDR arithmetic, and (2) that the computation can be physically laid out in two dimensions with only local flow of data between units, only limited memory within each unit, and only limited data flow to/from the host machine, in such a way that the computation makes efficient use of the machine's resources (area, time, power). The first requirement is referred to as "Accuracy" and the second requirement "Efficiency." Applications that meet both requirements running in this model will function well on many kinds of LPHDR machines, and thus those machines are a broadly useful invention.

**[0087]** Applications are tested using two embodiments for the machine's arithmetic. One uses accurate floating point arithmetic but multiplies the result of each arithmetic operation by a uniformly chosen random number between .99 and 1.01. In the following discussion, this embodiment is denoted "fp+noise". It may represent the results produced by an analog embodiment of the machine.

**[0088]** A second embodiment uses logarithmic arithmetic with a value representation as shown in FIG. 5. The arithmetic is repeatable, that is, not noisy, but because of the short fraction size it produces errors of up to approximately 1-2% in each operation. In the following discussion, this embodiment is denoted "lns". It may represent the results produced by a particular digital embodiment of the machine.

**[0089]** To demonstrate usefulness of embodiments of the invention, we shall discuss three computational tasks that are enabled by embodiments of the invention and which in turn enable a variety of practical applications. Two of the tasks are related to finding nearest neighbors and the other is related to processing visual information. We shall describe the tasks, note their practical application, and then demonstrate that each task is solvable using the general model described above and thus solvable using embodiments of the present

invention.

Application 1: Finding Nearest Neighbors

**[0090]** Given a large set of vectors, called Examples, and a given vector, called Test, the nearest neighbor problem ("NN") is to find the Example which is closest to Test where the distance metric is the square of the Euclidean distance (sum of squares of distances between respective components).

**[0091]** NN is a widely useful computation. One use is for data compression, where it is called "vector quantization". In this application we have a set of relatively long vectors in a "code book" (these are the Examples) and associated short code words (for instance the index of the vector in the code book). We move through a sequence of vectors to be compressed, and for each such vector (Test), find the nearest vector in the code book and output the corresponding code word. This reduces the sequence of vectors to the shorter sequence of code words. Because the code words do not completely specify the original sequence of vectors, this is a lossy form of data compression. Among other applications, it may be used in speech compression and in the MPEG standards.

**[0092]** Another application of NN would be in determining whether snippets of video occur in a large video database. Here we might abstract frames of video from the snippet into feature vectors, using known methods, such as color histograms, scale invariant feature extraction, etc. The Examples would be analogous feature vectors extracted from the video database. We would like to know whether any vector from the snippet was close to any vector from the database, which NN can help us decide.

**[0093]** In many applications of nearest neighbor, we would prefer to find the true nearest neighbor but it is acceptable if we sometimes find another neighbor that is only slightly farther away or if we almost always find the true nearest neighbor. Thus, an approximate solution to the nearest neighbor problem is useful, especially if it can be computed especially quickly, or at low power, or with some other advantage compared to an exact solution.

**[0094]** We shall now show that approximate nearest neighbor is

computable using embodiments of the present invention in a way that meets the criteria of Accuracy and Efficiency.

**[0095]**     Algorithm. The following describes an algorithm which may be performed by machines implemented according to embodiments of the present invention, such as by executing software including instructions for performing the algorithm. The inputs to the algorithm are a set of Examples and a Test vector. The algorithm seeks to find the nearest (or almost nearest) Example to the Test.

**[0096]**     In the simplest version of the algorithm, the number of Examples may be no larger than the number of PEs and each vector must be short enough to fit within a single PE's memory. The Examples are placed into the memories associated with the PEs, so that one Example is placed in each PE. Given a Test, the Test is passed through all the PEs, in turn. Accompanying the Test as it passes through the PEs is the distance from the Test to the nearest Example found so far, along with information that indicates what PE (and thus what Example) yielded that nearest Example found so far. Each PE computes the distance between the Test and the Example stored in that PE's memory, and then passes along the Test together with either the distance and indicator that was passed into this PE (if the distance computed by this PE exceeded the distance passed into the PE) or the distance this PE computed along with information indicating this PE's Example is the nearest so far (if the distance computed by this PE is less than the distance passed into the PE). Thus, the algorithm is doing a simple minimization operation as the Test is passed through the set of PEs. When the Test and associated information leave the last PE, the output is a representation of which PE (and Example) was closest to the Test, along with the distance between that Example and the Test.

**[0097]**     In a more efficient variant of this algorithm, the Test is first passed along, for example, the top row, then every column passes the Test and associated information downward, effectively doing a search in parallel with other columns, and once the information reaches the bottom it passes across the bottom row computing a minimum distance Example of all the columns processed so far as it passes across the row. This means that the time required



Attorney Docket No. A0006-1001C2

to process the Test is proportional to (the greater of) the number of PEs in a row or column.

**[0098]** An enhancement of this algorithm proceeds as above but computes and passes along information indicating both the nearest and the second nearest Example found so far. When this information exits the array of PEs, the digital processor that is hosting the PE array computes (in high precision) the distance between the Test and the two Examples indicated by the PE array, and the nearer of the two is output as the likely nearest neighbor to the Test.

**[0099]** Accuracy. We expressed the arithmetic performed by the enhanced algorithm described above as code in the C programming language. That code computes both nearest neighbors, which are discussed here, along with weighted scores, which are discussed below.

**[0100]** The C code performs the same set of arithmetic operations in the same order using the same methods of performing arithmetic as an actual implementation of the present invention, such as one implemented in hardware. It thus yields the same results as the enhanced algorithm would yield when running on an implementation of the present invention. (How the algorithm is organized to run efficiently on such an implementation is discussed below in the section on Efficiency.)

**[0101]** In particular, when computing the distance between the Test and each Example, the code uses Kahan's method, discussed below, to perform the possibly long summation required to form the sum of the squares of the distances between vector components of the Test and Example.

**[0102]** The C code contains several implementations for arithmetic, as discussed above. When compiled with "#define fp" the arithmetic is done using IEEE standard floating point. If a command line argument is passed in to enable noisy arithmetic, then random noise is added to the result of every calculation. This is the "fp+noise" form of arithmetic. When compiled without "#define fp" the arithmetic is done using low precision logarithmic arithmetic with a 6 bit base-2 fraction. This is the "lns" form of arithmetic.

Attorney Docket No. A0006-1001C2

**[0103]** When the code was run it produced traces showing the results of the computations it performed. These traces, shown below, show that with certain command line arguments the enhanced algorithm yielded certain results for LPHDR nearest neighbor calculations. These results provide details showing the usefulness of this approach. We shall discuss the results briefly here.

**[0104]** The first results are for "fp+noise". Ten distinct runs were performed. Each run generated one million random Example vectors of length five, where each component of each vector was drawn from  $N(0,1)$ —the Gaussian (normal) distribution with mean zero and standard deviation 1. Each run then generated one hundred Test vectors of length five, where each component of each vector also was drawn from  $N(0,1)$ . For each Test, the nearest neighbor was computed both according to the enhanced algorithm above and according to the standard nearest neighbor method using high precision floating point arithmetic. A count was kept of the number of times the enhanced algorithm yielded the same result as the standard floating point method. The results were as follows:

```
% ./a.out 5 10 1000000 100 1
```

Representation is Floating Point with noise.

Run 1. On 100 tests, 100(100.0%) matches and 0.81% mean score error.

Run 2. On 100 tests, 100(100.0%) matches and 0.84% mean score error.

Run 3. On 100 tests, 100(100.0%) matches and 0.98% mean score error.

Run 4. On 100 tests, 100(100.0%) matches and 0.81% mean score error.

Run 5. On 100 tests, 100(100.0%) matches and 0.94% mean score error.

Run 6. On 100 tests, 100(100.0%) matches and 0.82% mean score error.

Run 7. On 100 tests, 100(100.0%) matches and 0.78% mean score error.

Run 8. On 100 tests, 100(100.0%) matches and 0.86% mean score error.

Run 9. On 100 tests, 100(100.0%) matches and 0.85% mean score error.

Run 10. On 100 tests, 99(99.0%) matches and 0.86% mean score error.

Average percentage of time LPHDR (with final DP correction) finds nearest example = 99.90%.

Attorney Docket No. A0006-1001C2

Average score error between LPHDR and DP = 0.85%.

**[0105]** The "mean score error" values are considered below in the discussion of weighted scores. The "matches" information is relevant here.

**[0106]** Of the ten runs, only one had any test, of the 100 tests performed, which yielded a nearest neighbor different from what the usual high precision method yielded. Thus, the average percentage of matches between the enhanced algorithm running with "fp+noise" arithmetic and the usual method was 99.9%.

**[0107]** A similar computation was then performed using "lms" arithmetic. In this case, the results were:

% ./a.out 5 10 1000000 100 0

Representation is LNS without noise.

Run 1. On 100 tests, 100(100.0%) matches and 0.15% mean score error.

Run 2. On 100 tests, 100(100.0%) matches and 0.07% mean score error.

Run 3. On 100 tests, 100(100.0%) matches and 0.08% mean score error.

Run 4. On 100 tests, 100(100.0%) matches and 0.09% mean score error.

Run 5. On 100 tests, 100(100.0%) matches and 0.11% mean score error.

Run 6. On 100 tests, 100(100.0%) matches and 0.16% mean score error.

Run 7. On 100 tests, 100(100.0%) matches and 0.07% mean score error.

Run 8. On 100 tests, 100(100.0%) matches and 0.13% mean score error.

Run 9. On 100 tests, 99(99.0%) matches and 0.17% mean score error.

Run 10. On 100 tests, 98(98.0%) matches and 0.16% mean score error.

Average percentage of time LPHDR (with final DP correction) finds nearest example = 99.70%.

Average score error between LPHDR and DP = 0.12%.

**[0108]** The average percentage of matches was 99.7%, slightly worse than for "fp+noise".

**[0109]** The accuracy shown by the enhanced nearest neighbor

Attorney Docket No. A0006-1001C2

algorithm using two forms of LPHDR arithmetic is surprising. To perform many calculations sequentially with 1% error and yet produce a final result with less than 1% error may seem counter-intuitive. Nonetheless, the LPHDR arithmetic proves effective, and the accuracy shown is high enough to be useful in applications for which approximate nearest neighbor calculations are useful.

**[0110]** As an extreme case, a variant of fp+noise was tested in which the noise varied uniformly from +10% to -5%. Thus, each arithmetic operation produced a result that was between 10% too large and 5% too small. The enhanced nearest neighbor algorithm, as described above, was performed where each run generated 100,000 Example vectors. The surprising results, below, show that even with this extreme level of imprecise, noisy, and non-zero mean LPHDR arithmetic, useful results can be achieved.

Run 1. On 100 tests, 97(97.0%) matches.

Run 2. On 100 tests, 100(100.0%) matches.

Run 3. On 100 tests, 100(100.0%) matches.

Run 4. On 100 tests, 98(98.0%) matches.

Run 5. On 100 tests, 98(98.0%) matches.

Run 6. On 100 tests, 99(99.0%) matches.

Run 7. On 100 tests, 99(99.0%) matches.

Run 8. On 100 tests, 99(99.0%) matches.

Run 9. On 100 tests, 99(99.0%) matches.

Run 10. On 100 tests, 99(99.0%) matches.

Average percentage of time LPHDR (with final DP correction) finds nearest example = 98.80%.

**[0111]** Efficiency. In contrast to the surprising Accuracy results, it is clear to those having ordinary skill in the art that the calculations of the enhanced nearest neighbor algorithm can be performed efficiently in the computing model presented, where the arithmetic/memory units are connected in a two-dimensional physical layout, using only local communication between PEs.

Attorney Docket No. A0006-1001C2

However, this does not address the matter of keeping the machine busy doing useful work using only low bandwidth to the host machine.

**[0112]** When computing the nearest neighbor to a single Test, the Test flows across all the PEs in the array. As discussed above, if the array is an  $M \times M$  grid, it takes at least  $O(M)$  steps for the Test to pass through the machine and return results to the host. During this time the machine performs  $O(M \times M)$  nearest neighbor distance computations, but since the machine is capable of performing  $O(M \times M)$  calculations at each step, a factor of  $O(M)$  is lost.

**[0113]** This speedup, compared to a serial machine, of a factor of  $O(M)$  is significant and useful. However, the efficiency can be even higher. If sufficiently many Test vectors, say  $O(M)$ , or more, are to be processed then they can be streamed into the machine and made to flow through in a pipelined fashion. The time to process  $O(M)$  Tests remains  $O(M)$ , the same as for a single Test, but now the machine performs  $O(M) \times O(M \times M)$  distance computations, and thus within a constant factor the full computing capacity of the machine is used.

**[0114]** Thus, the machine is especially efficient if it is processing at least as many Test vectors as the square root of the number of PEs. There are applications that fit well into this form, such as pattern recognition or compression of many independent Tests (e.g., blocks of an image, parts of a file, price histories of independent stocks) as well as the problem of finding the nearest neighbor to every Example in the set of Examples. This is in contrast to the general view among those having ordinary skill in the art, as discussed above, that machines with very many arithmetic processing elements on a single chip, or similar, are not very useful.

#### Application 2: Distance Weighted Scoring

**[0115]** A task related to Nearest Neighbor is Distance Weighted Scoring. In this task, each Example has an associated Score. This is a number that in some way characterizes the Example. For instance, if the Examples are abstractions of the history of prices of a given stock, the Scores might be historical probabilities of whether the price is about to increase or decrease. Given a Test vector, the task is to form a weighted sum of the Scores of all the



Attorney Docket No. A0006-1001C2

Examples, where the weights are a diminishing function of the distance from the Test to the respective Examples. For example, this weighted score might be taken as a prediction of the future price of the stock whose history is represented by the Test. This use of embodiments of the invention might help support, for instance, high speed trading of stocks, as is performed by certain "quantitative" hedge funds, despite the general view by those having ordinary skill in the art that low precision computation is not of use in financial applications.

**[0116]** The C code described above computes weighted scores along with nearest neighbors. The scores assigned to Examples in this computation are random numbers drawn uniformly from the range  $[0,1]$ . The weight for each Example in this computation is defined to be the un-normalized weight for the Example divided by the sum of the un-normalized weights for all Examples, where the un-normalized weight for each Example is defined to be the reciprocal of the sum of one plus the squared distance from the Example to the Test vector. As discussed above, the code performs a number of runs, each producing many Examples and Tests, and compares results of traditional floating point computations with results calculated using fp+noise and Ins arithmetic.

**[0117]** Looking again at the trace results of running the simulation, above, we see that for fp+noise the LPHDR weighted scores on average were within .85% of the correct value and never were as much as 1% different. For Ins arithmetic the errors were even smaller, averaging just .12% error.

**[0118]** These results are surprising given that computing an overall weighted score involves summing the individual weighted scores associated with each Example. Since each run was processing 1,000,000 Examples, this means that the sums were over one million small positive values. The naive method of summing one million small values with errors of about 1% in each addition should yield results that approximate noise. However, the code performs its sums using a long known method invented by Kahan (Kahan, William (January 1965), "Further remarks on reducing truncation errors", Communications of the ACM 8 (1): 40). The method makes it feasible to perform long sums, such as are done for Distance Weighted Scores, or as might be used in computational finance

Attorney Docket No. A0006-1001C2

when computing prices of derivative securities using Monte Carlo methods, or for performing deconvolution in image processing algorithms, as will be discussed next.

**[0119]** The Efficiency of this algorithm is similar to that of NN, as discussed earlier. If many Test vectors are processed at once, the machine performs especially efficiently.

#### Application 3: Removing motion blur in images

**[0120]** In order to gather sufficient light to form an image, camera shutters are often left open for long enough that camera motion can cause blurring. This can happen as a result of camera shake in inexpensive consumer cameras as well as with very expensive but fast moving cameras mounted on satellites or aircraft. If the motion path of the camera is known (or can be computed) then the blur can be substantially removed using various deblurring algorithms. One such algorithm is the Richardson-Lucy method ("RL"), and we show here that embodiments of the present invention can run that algorithm and produce useful results. Following the discussion format above, we discuss criteria of Accuracy and Efficiency.

**[0121]** Algorithm. The Richardson-Lucy algorithm is well known and widely available. Assume that an image has been blurred using a known kernel. In particular, assume that the kernel is a straight line and that the image has been oriented so that the blur has occurred purely in a horizontal direction. Consider the particular kernel for which the J'th pixel in each row of the blurred image is the uniformly weighted mean of pixels J through J+31 in the original unblurred image.

**[0122]** Accuracy. We implemented in the C programming language a straightforward version of the RL method that uses LPHDR arithmetic. The program reads a test image, blurs it using the kernel discussed above, then deblurs it using either fp+noise or lns arithmetic. The RL algorithm computes sums, such as when convolving the kernel with the current approximation of the deblurred image. Our implementation computes these sums using the Kahan method, discussed earlier. FIG. 7 shows the test image in original form. It is a

Attorney Docket No. A0006-1001C2

satellite picture of a building used during Barack Obama's inauguration. FIG. 8 shows the image extremely blurred by the kernel. It is difficult to see any particular objects in this image. FIG. 9 shows the result of deblurring using standard floating point arithmetic. FIG. 10 shows the result of deblurring using fp+noise arithmetic, and FIG.11 shows the result of deblurring using lns arithmetic. In all these cases the image is sufficiently restored that it is possible to recognize buildings, streets, parking lots, and cars.

**[0123]** In addition to displaying the images herein for judgement using the human eye, we computed a numerical measure of deblurring performance. We computed the mean difference, over all pixels in the image, between each original pixel value (a gray scale value from 0 to 255) and the corresponding value in the image reconstructed by the RL method. Those numerical measures are shown below in Table 1:

<u>Image type</u>	<u>Mean pixel error</u>
Blurred	32.0
RL using standard floating point	13.0
RL using fp+noise	13.8
RL using lns	14.8

**Table 1**

**[0124]** These results, together with the subjective but important judgements made by the human eye, show that LPHDR arithmetic provides a substantial and useful degree of deblurring compared to standard floating point arithmetic. Further, in this example we chose an extreme degree of blurring, to better convey the concept and visual impact of the deblurring using LPHDR arithmetic. On more gentle and typical blur kernels, the resulting deblurred images are much closer to the originals than in this case, as can be seen by shrinking the kernel length and running the RL algorithm with LPHDR arithmetic on those more typical cases.

**[0125]** Efficiency. It is clear to those with ordinary skill in the art that Richardson-Lucy using a local kernel performs only local computational operations. An image to be deblurred can be loaded into the PE array, storing

Attorney Docket No. A0006-1001C2

one or more pixels per PE, the deconvolution operation of RL can then be iterated dozens or hundreds of times, and the deblurred image can be read back to the host processor. As long as sufficient iterations are performed, this makes efficient use of the machine.

**[0126]** An extreme form of image deblurring is the Iterative Reconstruction method used in computational tomography. Reconstructing 3D volumes from 2D projections is an extremely computational task. The method discussed above generalizes naturally to Iterative Reconstruction and makes efficient use of the machine.

**[0127]** Among the advantages of embodiments of the invention are one or more of the following.

**[0128]** PEs implemented according to certain embodiments of the present invention may be relatively small for PEs that can do arithmetic. This means that there are many PEs per unit of resource (e.g., transistor, area, volume), which in turn means that there is a large amount of arithmetic computational power per unit of resource. This enables larger problems to be solved with a given amount of resource than does traditional computer designs. For instance, a digital embodiment of the present invention built as a large silicon chip fabricated with current state of the art technology might perform tens of thousand of arithmetic operations per cycle, as opposed to hundreds in a conventional GPU or a handful in a conventional multicore CPU. These ratios reflect an architectural advantage of embodiments of the present invention that should persist as fabrication technology continues to improve, even as we reach nanotechnology or other implementations for digital and analog computing.

**[0129]** Doing arithmetic with few resources generally means, and in the embodiments shown specifically means, that the arithmetic is done using low power. As a result, a machine implemented in accordance with embodiments of the present invention can have extremely high performance with reasonable power (for instance in the tens of watts) or low power (for instance a fraction of a watt) with reasonably high performance. This means that such embodiments may be suitable for the full range of computing, from supercomputers, through

Attorney Docket No. A0006-1001C2

desktops, down to mobile computing. Similarly, and since cost is generally associated with the amount of available resources, embodiments of the present invention may provide a relatively high amount of computing power per unit of cost compared to conventional computing devices.

**[0130]** The SIMD architecture is rather old and is frequently discarded as an approach to computer design by those having ordinary skill in the art. However, if the processing elements of a SIMD machine can be made particularly small while retaining important functionality, such as general arithmetic ability, the architecture can be useful. The embodiments presented herein have precisely those qualities.

**[0131]** The discovery that massive amounts of LPHDR arithmetic is useful as a fairly general computing framework, as opposed to the common belief that it is not useful, can be an advantage in any (massively or non-massively) parallel machine design or non-parallel design, not just in SIMD embodiments. It could be used in FPGAs, FPAAs, GPU/SIMT machines, MIMD machines, and in any kind of machine that uses compact arithmetic processing elements to perform large amounts of computation using a small amount of resources (like transistors or volume).

**[0132]** Another advantage of embodiments of the present invention is that they are not merely useful for performing computations efficiently in general, but that they can be used to tackle a variety of real-world problems which are typically assumed to require high-precision computing elements, even though such embodiments include only (or predominantly) low-precision computing elements. Although several examples of such real-world problems have been presented herein, and although we have also had success implementing non-bonded force field computations for molecular dynamics simulation and other tasks, these are merely examples and do not constitute an exhaustive set of the real-world problems that embodiments of the present invention may be used to solve.

**[0133]** The embodiments disclosed above are merely examples and do not constitute limitations of the present invention. Rather, embodiments of the



Attorney Docket No. A0006-1001C2

present invention may be implemented in a variety of other ways, such as the following.

**[0134]** For example, embodiments of the present invention may represent values in any of a variety of ways, such as by using digital or analog representations, such as fixed point, logarithmic, or floating point representations, voltages, currents, charges, pulse width, pulse density, frequency, probability, spikes, timing, or combinations thereof. These underlying representations may be used individually or in combination to represent the LPHDR values. LPHDR arithmetic circuits may be implemented in any of a variety of ways, such as by using various digital methods (which may be parallel or serial, pipelined or not) or analog methods or combinations thereof. Arithmetic elements may be connected using various connection architectures, such as nearest 4, nearest 8, hops of varying degree, and architectures which may or may not be rectangular or grid-like. Any method may be used for communication among arithmetic elements, such as parallel or serial, digital or analog or mixed-mode communication. Arithmetic elements may operate synchronously or asynchronously, and may operate globally simultaneously or not. Arithmetic elements may be implemented, for example, on a single physical device, such as a silicon chip, or spread across multiple devices and an embodiment built from multiple devices may have its arithmetic elements connected in a variety of ways, including for example being connected as a grid, torus, hypercube, tree, or other method. Arithmetic elements may be connected to a host machine, if any, in a variety of ways, depending on the cost and bandwidth and other requirements of a particular embodiment. For example there may be many host machines connected to the collection of arithmetic elements.

**[0135]** Although certain embodiments of the present invention are described as being implemented as a SIMD architecture, this is merely an example and does not constitute a limitation of the present invention. For example, embodiments of the present invention may be implemented as reconfigurable architectures, such as but not limited to programmable logic devices, field programmable analog arrays, or field programmable gate array

architectures, such as a design in which existing multiplier blocks of an FPGA are replaced with or supplemented by LPHDR arithmetic elements of any of the kinds disclosed herein, or for example in which LPHDR elements are included in a new or existing reconfigurable device design. As another example, embodiments of the present invention may be implemented as a GPU or SIMT-style architecture which incorporates LPHDR arithmetic elements of any of the kinds disclosed herein. For example, LPHDR elements could supplement or replace traditional arithmetic elements in current or new graphics processing unit designs. As yet another example, embodiments of the present invention may be implemented as a MIMD-style architecture which incorporates LPHDR arithmetic elements of any of the kinds disclosed herein. For example, LPHDR arithmetic elements could supplement or replace traditional arithmetic elements in current or new MIMD computing system designs. As yet another example, embodiments of the present invention may be implemented as any kind of machine, including a massively parallel machine, which uses compact arithmetic processing elements to provide large amounts of arithmetic computing capability using a small amount of resources (for example, transistors or area or volume) compared with traditional architectures.

**[0136]** Although certain embodiments of the present invention are described herein as executing software, this is merely an example and does not constitute a limitation of the present invention. Alternatively, for example, embodiments of the present invention may be implemented using microcode or a hardware sequencer or state machine or other controller to control LPHDR arithmetic elements of any of the kinds disclosed herein. Alternatively, for example, embodiments of the present invention may be implemented using hardwired, burned, or otherwise pre-programmed controllers to control LPHDR arithmetic elements of any of the kinds disclosed herein.

**[0137]** Although certain embodiments of the present invention are described herein as being implemented using custom silicon as the hardware, this is merely an example and does not constitute a limitation of the present invention. Alternatively, for example, embodiments of the present invention may

be implemented using FPGA or other reconfigurable chips as the underlying hardware, in which the FPGAs or other reconfigurable chips are configured to perform the LPHDR operations disclosed herein. As another example, embodiments of the present invention may be implemented using any programmable conventional digital or analog computing architecture (including those which use high-precision computing elements, including those which use other kinds of non-LPHDR hardware to perform LPHDR arithmetic, and including those which are massively parallel) which has been programmed with software to perform the LPHDR operations disclosed herein. For example, embodiments of the present invention may be implemented using a software emulator of the functions disclosed herein.

**[0138]** As yet another example, embodiments of the present invention may be implemented using 3D fabrication technologies, whether based on silicon chips or otherwise. Some example embodiments are those in which a memory chip has been bonded onto a processor or other device chip or in which several memory and/or processor or other device chips have been bonded to each other in a stack. 3D embodiments of the present invention are very useful as they may be denser than 2D embodiments and may enable 3D communication of information between the processing units, which enables more algorithms to run efficiently on those embodiments compared to 2D embodiments.

**[0139]** Although certain embodiments of the present invention are described herein as being implemented using silicon chip fabrication technology, this is merely an example and does not constitute a limitation of the present invention. Alternatively, for example, embodiments of the present invention may be implemented using technologies that may enable other sorts of traditional digital and analog computing processors or other devices. Examples of such technologies include various nanomechanical and nanoelectronic technologies, chemistry based technologies such as for DNA computing, nanowire and nanotube based technologies, optical technologies, mechanical technologies, biological technologies, and other technologies whether based on transistors or not that are capable of implementing LPHDR architectures of the kinds disclosed

Attorney Docket No. A0006-1001C2

herein.

**[0140]** Certain embodiments of the present invention have been described as “massively parallel” embodiments. Although certain embodiments of the present invention may include thousands, millions, or more arithmetic units, embodiments of the present invention may include any number of arithmetic units (as few as one). For example, even an embodiment which includes only a single LPHDR unit may be used within a serial processing unit or other device to provide a significant amount of LPHDR processing power in a small, inexpensive processor or other device.

**[0141]** For certain embodiments of the present invention, even if implemented using only digital techniques, the arithmetic operations may not yield deterministic, repeatable, or the most accurate possible results within the chosen low precision representation. For instance, on certain specific input values, an arithmetic operation may produce a result which is not the nearest value in the chosen representation to the true arithmetic result.

**[0142]** The degree of precision of a “low precision, high dynamic range” arithmetic element may vary from implementation to implementation. For example, in certain embodiments, a LPHDR arithmetic element produces results which include fractions, that is, values greater than zero and less than one. For example, in certain embodiments, a LPHDR arithmetic element produces results which are sometimes (or all of the time) no closer than 0.05% to the correct result (that is, the absolute value of the difference between the produced result and the correct result is no more than one-twentieth of one percent of the absolute value of the correct result). As another example, a LPHDR arithmetic element may produce results which are sometimes (or all of the time) no closer than 0.1% to the correct result. As another example, a LPHDR arithmetic element may produce results which are sometimes (or all of the time) no closer than 0.2% to the correct result. As yet another example, a LPHDR arithmetic element may produce results which are sometimes (or all of the time) no closer than 0.5% to the correct result. As yet further examples, a LPHDR arithmetic element may produce results which are sometimes (or all of the time) no closer than 1%, or

Attorney Docket No. A0006-1001C2

2%, or 5%, or 10%, or 20% to the correct result.

**[0143]** Besides having various possible degrees of precision, implementations may vary in the dynamic range of the space of values they process. For example, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range approximately from one millionth to one million. As another example, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range approximately from one billionth to one billion. As yet another example, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range approximately from one sixty five thousandth to sixty five thousand. As yet further examples, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range from any specific value between zero and one sixty five thousandth up to any specific value greater than sixty five thousand. As yet further examples, other embodiments may process values in spaces with dynamic ranges that may combine and may fall between the prior examples, for example ranging from approximately one billionth to ten million. In all of these example embodiments of the present invention, as well as in other embodiments, the values that we are discussing may be signed, so that the above descriptions characterize the absolute values of the numbers being discussed.

**[0144]** The frequency with which LPHDR arithmetic elements may yield only approximations to correct results may vary from implementation to implementation. For example, consider an embodiment in which LPHDR arithmetic elements can perform one or more operations (perhaps including, for example, trigonometric functions), and for each operation the LPHDR elements each accept a set of inputs drawn from a range of valid values, and for each specific set of input values the LPHDR elements each produce one or more output values (for example, simultaneously computing both sin and cos of an input), and the output values produced for a specific set of inputs may be deterministic or non-deterministic. In such an example embodiment, consider further a fraction  $F$  of the valid inputs and a relative error amount  $E$  by which the



result calculated by an LPHDR element may differ from the mathematically correct result. In certain embodiments of the present invention, for each LPHDR arithmetic element, for at least one operation that the LPHDR unit is capable of performing, for at least fraction  $F$  of the possible valid inputs to that operation, for at least one output signal produced by that operation, the statistical mean, over repeated execution, of the numerical values represented by that output signal of the LPHDR unit, when executing that operation on each of those respective inputs, differs by at least  $E$  from the result of an exact mathematical calculation of the operation on those same input values, where  $F$  is 1% and  $E$  is 0.05%. In several other example embodiments,  $F$  is not 1% but instead is one of 2%, or 5%, or 10%, or 20%, or 50%. For each of these example embodiments, each with some specific value for  $F$ , there are other example embodiments in which  $E$  is not 0.05% but instead is 0.1%, or 0.2%, or 0.5%, or 1%, or 2%, or 5%, or 10%, or 20%. These varied embodiments are merely examples and do not constitute limitations of the present invention.

**[0145]** For certain devices (such as computers or processors or other devices) embodied according the present invention, the number of LPHDR arithmetic elements in the device (e.g., computer or processor or other device) exceeds the number, possibly zero, of arithmetic elements in the device which are designed to perform high dynamic range arithmetic of traditional precision (that is, floating point arithmetic with a word length of 32 or more bits). If  $N_L$  is the total number of LPHDR elements in such a device, and  $N_H$  is the total number of elements in the device which are designed to perform high dynamic range arithmetic of traditional precision, then  $N_L$  exceeds  $T(N_H)$ , where  $T()$  is some function. Any of a variety of functions may be used as the function  $T()$ . For example, in certain embodiments,  $T(N_H)$  may be twenty plus three times  $N_H$ , and the number of LPHDR arithmetic elements in the device may exceed twenty more than three times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed fifty more than five times the number of

Attorney Docket No. A0006-1001C2

arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As yet another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed one hundred more than five times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As yet another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed one thousand more than five times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As yet another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed five thousand more than five times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. Certain embodiments of the present invention may be implemented within a single physical device, such as but not limited to a silicon chip or a chip stack or a chip package or a circuit board, and the number NL of LPHDR elements in the physical device and the number NH of elements designed to perform high dynamic range arithmetic of traditional precision in the physical device may be the total counts of the respective elements within that physical device. Certain embodiments of the present invention may be implemented in a computing system including more than one physical device, such as but not limited to a collection of silicon chips or chip stacks or chip packages or circuit boards coupled to and communicating with each other using any means (such as a bus, switch, any kind of network connection, or other means of communication), and in this case the number NL of LPHDR elements in the computing system and the number NH of elements designed to perform high dynamic range arithmetic of traditional precision in the computing system may be the total counts of the respective elements within all those physical devices jointly.

**[0146]** Certain embodiments of the present invention may constitute, or may be part of, processors, which are devices capable of executing software to perform computations. Such processors may include mechanisms for storing

Attorney Docket No. A0006-1001C2

software, for using the software to determine what operations to perform, for performing those operations, for storing numerical data, for modifying data according to the software specified operations, and for communicating with devices connected to the processor. Processors may be reconfigurable devices, such as, without limitation, field programmable arrays. Processors may be co-processors to assist host machines or may be capable of operating independently of an external host. Processors may be formed as a collection of component host processors and co-processors of various types, such as CPUs, GPUs, FPGAs, or other processors or other devices, which in the art may be referred to as a heterogeneous processor design or heterogeneous computing system, some or all of which components might incorporate the same or distinct varieties of embodiments of the present invention.

**[0147]** Embodiments of the present invention may, however, be implemented in devices in addition to or other than processors. For example, a computer including a processor and other components (such as memory coupled to the processor by a data path), wherein the processor includes components for performing LPHDR operations in any of the ways disclosed herein, is an example of an embodiment of the present invention. More generally, any device or combination of devices, whether or not falling within the meaning of a “processor,” which performs the functions disclosed herein may constitute an example of an embodiment of the present invention.

**[0148]** More generally, any of the techniques described above may be implemented, for example, in hardware, software tangibly stored on a computer-readable medium, firmware, or any combination thereof. The techniques described above may be implemented in one or more computer programs executing on a programmable computer including a processor, a storage medium readable by the processor (including, for example, volatile and non-volatile memory and/or storage elements), at least one input device, and at least one output device. Program code may be applied to input entered using the input device to perform the functions described and to generate output. The output may be provided to one or more output devices.

**[0149]** Each computer program within the scope of the claims below may be implemented in any programming language, such as assembly language, machine language, a high-level procedural programming language, or an object-oriented programming language. The programming language may, for example, be a compiled or interpreted programming language.

**[0150]** Each such computer program may be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a computer processor. Method steps of the invention may be performed by a computer processor executing a program tangibly embodied on a computer-readable medium to perform functions of the invention by operating on input and generating output. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, the processor receives instructions and data from a read-only memory and/or a random access memory. Storage devices suitable for tangibly embodying computer program instructions include, for example, all forms of non-volatile memory, such as semiconductor memory devices, including EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROMs. Any of the foregoing may be supplemented by, or incorporated in, specially-designed ASICs (application-specific integrated circuits) or FPGAs (Field-Programmable Gate Arrays). A computer can generally also receive programs and data from a storage medium such as an internal disk (not shown) or a removable disk. These elements will also be found in a conventional desktop or workstation computer as well as other computers suitable for executing computer programs implementing the methods described herein, which may be used in conjunction with any digital print engine or marking engine, display monitor, or other raster output device capable of producing color or gray scale pixels on paper, film, display screen, or other output medium.

### **Claims**

1. A device comprising:

at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,

wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/65,000$  through  $65,000$  and for at least  $X=5\%$  of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least  $X\%$  of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least  $Y=.05\%$  from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and

at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit.

2. The method of claim 1, wherein the at least one first computing device comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine.

3. The device of claim 2, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.

4. The device of claim 3, wherein  $X=10\%$ .



Attorney Docket No. A0006-1001C2

5. The device of claim 3, wherein  $Y=.2\%$ .

6. The device of claim 3, wherein  $X=10\%$  and  $Y=.2\%$ .

7. The device of claim 3, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/1,000,000$  through  $1,000,000$ .

8. The device of claim 3, wherein the first operation is multiplication.

9. A device comprising:

at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,

wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/65,000$  through  $65,000$  and for at least  $X=5\%$  of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least  $X\%$  of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least  $Y=.05\%$  from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and

wherein the at least one first LPHDR execution unit comprises at least one of a chemistry-technology-based execution unit, a biological-technology-based execution unit, a DNA-technology-based execution unit, a nanomechanical-technology-based execution unit, a nanoelectronic-technology-based execution unit, a nanowire-technology-based execution unit, a nanotube-technology-based execution unit, and an optical-technology-based execution unit.

Attorney Docket No. A0006-1001C2

10. The device of claim 9, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.

11. The device of claim 10, wherein  $X=10\%$ .

12. The device of claim 10, wherein  $Y=.2\%$ .

13. The device of claim 10, wherein  $X=10\%$  and  $Y=.2\%$ .

14. The device of claim 10, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/1,000,000$  through  $1,000,000$ .

15. The device of claim 10, wherein the first operation is multiplication.

16. A device comprising:

a plurality of components comprising:

at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,

wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/65,000$  through  $65,000$  and for at least  $X=5\%$  of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least  $X\%$  of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing

Attorney Docket No. A0006-1001C2

the first operation on that input differs by at least  $Y=.05\%$  from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.

17. The device of claim 16, wherein the plurality of components are arranged in a stack.

18. The device of claim 17, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.

19. The device of claim 18, wherein  $X=10\%$ .

20. The device of claim 18, wherein  $Y=.2\%$ .

21. The device of claim 18, wherein  $X=10\%$  and  $Y=.2\%$ .

22. The device of claim 18, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/1,000,000$  through  $1,000,000$ .

23. The device of claim 18, wherein the first operation is multiplication.

24. The device of claim 16, wherein the plurality of components are bonded.

25. The device of claim 16, wherein the plurality of components are arranged in a stack and bonded.

Attorney Docket No. A0006-1001C2

26. A device comprising a computer processor and a computer-readable memory storing computer program instructions, wherein the computer program instructions are executable by the processor to emulate a second device comprising:

at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value;

wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/65,000$  through  $65,000$  and for at least  $X=5\%$  of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least  $X\%$  of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least  $Y=.05\%$  from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and

wherein the at least one first LPHDR execution unit comprises at least one of a chemistry-technology-based execution unit, a biological-technology-based execution unit, a DNA-technology-based execution unit, a nanomechanical-technology-based execution unit, a nanoelectronic-technology-based execution unit, a nanowire-technology-based execution unit, a nanotube-technology-based execution unit, and an optical-technology-based execution unit.

27. The device of claim 26, wherein the number of LPHDR execution units in the second device exceeds by at least one hundred the non-negative integer number of execution units in the second device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.

28. The device of claim 27, wherein  $X=10\%$ .

Attorney Docket No. A0006-1001C2

29. The device of claim 27, wherein  $Y=.2\%$ .

30. The device of claim 27, wherein  $X=10\%$  and  $Y=.2\%$ .

31. The device of claim 27, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/1,000,000$  through  $1,000,000$ .

32. The device of claim 27, wherein the first operation is multiplication.

33. A device comprising a computer processor and a computer-readable memory storing computer program instructions, wherein the computer program instructions are executable by the processor to emulate a second device comprising:

- a plurality of components comprising:

- at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value;

- wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/65,000$  through  $65,000$  and for at least  $X=5\%$  of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least  $X\%$  of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least  $Y=.05\%$  from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.



Attorney Docket No. A0006-1001C2

34. The device of claim 33, wherein the plurality of components are arranged in a stack.

35. The device of claim 34, wherein the number of LPHDR execution units in the second device exceeds by at least one hundred the non-negative integer number of execution units in the second device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.

36. The device of claim 35, wherein  $X=10\%$ .

37. The device of claim 35, wherein  $Y=.2\%$ .

38. The device of claim 35, wherein  $X=10\%$  and  $Y=.2\%$ .

39. The device of claim 35, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/1,000,000$  through  $1,000,000$ .

40. The device of claim 35, wherein the first operation is multiplication.

41. The device of claim 33, wherein the plurality of components are bonded.

42. The device of claim 33, wherein the plurality of components are arranged in a stack and bonded.

**Abstract**

A processor or other device, such as a programmable and/or massively parallel processor or other device, includes processing elements designed to perform arithmetic operations (possibly but not necessarily including, for example, one or more of addition, multiplication, subtraction, and division) on numerical values of low precision but high dynamic range ("LPHDR arithmetic"). Such a processor or other device may, for example, be implemented on a single chip. Whether or not implemented on a single chip, the number of LPHDR arithmetic elements in the processor or other device in certain embodiments of the present invention significantly exceeds (e.g., by at least 20 more than three times) the number of arithmetic elements, if any, in the processor or other device which are designed to perform high dynamic range arithmetic of traditional precision (such as 32 bit or 64 bit floating point arithmetic).

## Electronic Patent Application Fee Transmittal

<b>Application Number:</b>	13849606			
<b>Filing Date:</b>	25-Mar-2013			
<b>Title of Invention:</b>	Processing with Compact Arithmetic Processing Element			
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates			
<b>Filer:</b>	Robert Plotkin/Karen Del Greco			
<b>Attorney Docket Number:</b>	A0006-1001C2			
Filed as Small Entity				
<b>Utility under 35 USC 111(a) Filing Fees</b>				
<b>Description</b>	<b>Fee Code</b>	<b>Quantity</b>	<b>Amount</b>	<b>Sub-Total in USD(\$)</b>
<b>Basic Filing:</b>				
Utility filing Fee (Electronic filing)	4011	1	70	70
Utility Search Fee	2111	1	300	300
Utility Examination Fee	2311	1	360	360
<b>Pages:</b>				
<b>Claims:</b>				
Claims in excess of 20	2202	22	40	880
Independent Claims in Excess of 3	2201	2	210	420
<b>Miscellaneous-Filing:</b>				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Late Filing Fee for Oath or Declaration	2051	1	70	70
<b>Petition:</b>				
<b>Patent-Appeals-and-Interference:</b>				
<b>Post-Allowance-and-Post-Issuance:</b>				
<b>Extension-of-Time:</b>				
<b>Miscellaneous:</b>				
<b>Total in USD (\$)</b>				<b>2100</b>

**Electronic Acknowledgement Receipt**

<b>EFS ID:</b>	16456459
<b>Application Number:</b>	13849606
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	6059
<b>Title of Invention:</b>	Processing with Compact Arithmetic Processing Element
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates
<b>Customer Number:</b>	24208
<b>Filer:</b>	Robert Plotkin/Karen Del Greco
<b>Filer Authorized By:</b>	Robert Plotkin
<b>Attorney Docket Number:</b>	A0006-1001C2
<b>Receipt Date:</b>	30-JUL-2013
<b>Filing Date:</b>	25-MAR-2013
<b>Time Stamp:</b>	16:30:15
<b>Application Type:</b>	Utility under 35 USC 111(a)

**Payment information:**

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	\$2100
RAM confirmation Number	3662
Deposit Account	501797
Authorized User	DEL GRECO, KAREN

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)



Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

**File Listing:**

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	Response_NoticeToFileMissing Parts_A0006-1001C2.pdf	52506 6dafc9b5cf97209a0883933ee09551e5a9ecce92	no	2

**Warnings:****Information:**

2	Specification	Specification.pdf	217473 e1bdf778cb286b890cd8543dfa80f5ae56ef3d8	no	58
---	---------------	-------------------	---	----	----

**Warnings:****Information:**

3	Fee Worksheet (SB06)	fee-info.pdf	40090 7f0b4906e8eac830b739d2435e04b0b281cb6bd3	no	2
---	----------------------	--------------	---	----	---

**Warnings:****Information:**

<b>Total Files Size (in bytes):</b>	310069
-------------------------------------	--------

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

**New Applications Under 35 U.S.C. 111**

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

**National Stage of an International Application under 35 U.S.C. 371**

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

**New International Application Filed with the USPTO as a Receiving Office**

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

<b>PATENT APPLICATION FEE DETERMINATION RECORD</b>						Application or Docket Number 13/849,606			
Substitute for Form PTO-875									
<b>APPLICATION AS FILED - PART I</b>									
(Column 1)		(Column 2)		SMALL ENTITY		OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA	RATE(\$)	FEE(\$)		RATE(\$)	FEE(\$)		
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	70		N/A			
SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A	N/A	300		N/A			
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	360		N/A			
TOTAL CLAIMS (37 CFR 1.16(j))	42	minus 20 = *	22	x 40 =	880				
INDEPENDENT CLAIMS (37 CFR 1.16(h))	5	minus 3 = *	2	x 210 =	420				
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$310 (\$155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).				0.00				
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))					0.00				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL		2030	TOTAL			
<b>APPLICATION AS AMENDED - PART II</b>									
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY		OTHER THAN SMALL ENTITY	
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)		
	Total (37 CFR 1.16(i))	*	Minus **	**	=	x	=		
	Independent (37 CFR 1.16(h))	*	Minus ***	***	=	x	=		
	Application Size Fee (37 CFR 1.16(s))								
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
TOTAL ADD'L FEE					TOTAL ADD'L FEE				
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE(\$)	ADDITIONAL FEE(\$)	RATE(\$)	ADDITIONAL FEE(\$)		
	Total (37 CFR 1.16(i))	*	Minus **	**	=	x	=		
	Independent (37 CFR 1.16(h))	*	Minus ***	***	=	x	=		
	Application Size Fee (37 CFR 1.16(s))								
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))								
TOTAL ADD'L FEE					TOTAL ADD'L FEE				
* If the entry in column 1 is less than the entry in column 2, write "0" in column 3. ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20". *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3". The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.									



## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
**United States Patent and Trademark Office**  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY. DOCKET NO	TOT CLAIMS	IND CLAIMS
13/849,606	03/25/2013	2193	0.00	A0006-1001C2	42	5

CONFIRMATION NO. 6059

24208  
 ROBERT PLOTKIN, PC  
 15 New England Executive Office Park  
 Burlington, MA 01803

## FILING RECEIPT



\*OC000000061393662\*

Date Mailed: 05/30/2013

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

## Inventor(s)

Joseph Bates, Newton, MA;

## Applicant(s)

Singular Computing LLC, Newton, MA

## Assignment For Published Patent Application

Singular Computing LLC, Newton, MA

**Power of Attorney:** The patent practitioners associated with Customer Number 24208

## Domestic Priority data as claimed by applicant

This application is a CON of 13/399,884 02/17/2012 PAT 8407273  
 which is a CON of 12/816,201 06/15/2010 PAT 8150902  
 which claims benefit of 61/218,691 06/19/2009

**Foreign Applications** for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see <http://www.uspto.gov> for more information.) - None.

*Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.*

Permission to Access - A proper **Authorization to Permit Access to Application by Participating Offices** (PTO/SB/39 or its equivalent) has been received by the USPTO.

**Projected Publication Date:** To Be Determined - pending completion of Missing Parts

**Non-Publication Request:** No

**Early Publication Request:** No

**\*\* SMALL ENTITY \*\***

**Title**

Processing with Compact Arithmetic Processing Element

**Preliminary Class**

708

**Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No**

**PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES**

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at <http://www.uspto.gov/web/offices/pac/doc/general/index.html>.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, <http://www.stopfakes.gov>. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

**LICENSE FOR FOREIGN FILING UNDER**  
**Title 35, United States Code, Section 184**  
**Title 37, Code of Federal Regulations, 5.11 & 5.15**

**GRANTED**

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

**NOT GRANTED**

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

---

***SelectUSA***

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit <http://www.SelectUSA.gov> or call +1-202-482-6800.





## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
 United States Patent and Trademark Office  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/849,606	03/25/2013	Joseph Bates	A0006-1001C2

CONFIRMATION NO. 6059

## FORMALITIES LETTER

24208  
 ROBERT PLOTKIN, PC  
 15 New England Executive Office Park  
 Burlington, MA 01803



Date Mailed: 05/30/2013

## NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

*Filing Date Granted***Items Required To Avoid Abandonment:**

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.  
*Applicant must submit \$70 to complete the basic filing fee for a small entity.*

The application is informal since it does not comply with the regulations for the reason(s) indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- A substitute specification in compliance with 37 CFR 1.52, 1.121(b)(3), and 1.125, is required. The substitute specification must be submitted with markings and be accompanied by a clean version (without markings) as set forth in 37 CFR 1.125(c) and a statement that the substitute specification contains no new matter (see 37 CFR 1.125(b)). The specification, claims, and/or abstract page(s) submitted is not acceptable and cannot be scanned or properly stored because:
  - The line spacing on the specification, claims, and/or abstract is not 1½ or double spaced (see 37 CFR 1.52(b)).

Applicant is cautioned that correction of the above items may cause the specification and drawings page count to exceed 100 pages. If the specification and drawings exceed 100 pages, applicant will need to submit the required application size fee.

The applicant needs to satisfy supplemental fees problems indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- Additional claim fees of \$ **1300** as a small entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.

- A surcharge (for late submission of the basic filing fee, search fee, examination fee or inventor's oath or declaration) as set forth in 37 CFR 1.16(f) of \$ **70** for a small entity in compliance with 37 CFR 1.27, must be submitted.

**SUMMARY OF FEES DUE:**

Total fee(s) required within **TWO MONTHS** from the date of this Notice is \$ **2100** for a small entity

- \$ **70** Statutory basic filing fee.
- \$ **70** Surcharge.
- The application search fee has not been paid. Applicant must submit \$ **300** to complete the search fee.
- The application examination fee has not been paid. Applicant must submit \$ **360** to complete the examination fee for a small entity in compliance with 37 CFR 1.27.
- Total additional claim fee(s) for this application is \$ **1300**
  - \$ **420** for **2** independent claims over 3.
  - \$ **880** for **22** total claims over 20.

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts  
Commissioner for Patents  
P.O. Box 1450  
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web.

<https://portal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html>

For more information about EFS-Web please call the USPTO Electronic Business Center at **1-866-217-9197** or visit our website at <http://www.uspto.gov/ebc>.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

/mhteklu/

---

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101



## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
**United States Patent and Trademark Office**  
 Address: COMMISSIONER FOR PATENTS  
 P.O. Box 1450  
 Alexandria, Virginia 22313-1450  
 www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
13/849,606	03/25/2013	Joseph Bates	A0006-1001C2

**CONFIRMATION NO. 6059****POA ACCEPTANCE LETTER**

24208  
 ROBERT PLOTKIN, PC  
 15 New England Executive Office Park  
 Burlington, MA 01803



\*OC000000061393422\*

Date Mailed: 05/30/2013

**NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY**

This is in response to the Power of Attorney filed 04/08/2013.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/dnguyen/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

## Declaration

*Equivalent to USPTO Form PTO/AIA/01*

Application Number: 13/849,606 Attorney Docket Number: A0006-1001C2  
Filing Date: 03/25/2013 Examiner Name: Michael D. Yaary  
First Named Inventor: BATES, Joseph Art Unit: N/A  
Title of Invention: Processing with Compact Arithmetic Processing Element

Each below-named inventor hereby declares that:

### 1. Application

This Declaration is directed to:

- ☐ the attached application; or  
☒ United States application or PCT international application number 13/849,606  
filed on 03/25/2013 (mm/dd/yyyy).

### 2. Declaration

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this Declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

### 3. Duty of Disclosure

I hereby acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56, including for continuation-in-part applications, material information which become available between the filing date of the prior application and the national or PCT international filing date of the continuation-in-part application.

4. Warning

The undersigned is/are cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. The undersigned is/are advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.



5. Signature(s)

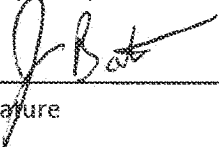
a. Inventor 1

Given Name (First and Middle):

Joseph

Family (Last) Name:

Bates

  
Signature

29 MARCH 2013

Date

b. Inventor 2

Given Name (First and Middle):

Family (Last) Name:

Signature

Date

c. Inventor 3

Given Name (First and Middle):

Family (Last) Name:

Signature

Date

d. Inventor 4

Given Name (First and Middle):

Family (Last) Name:

Signature

Date

e. Inventor 5

Given Name (First and Middle):

\_\_\_\_\_

Family (Last) Name:

\_\_\_\_\_

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Date

f. Inventor 6

Given Name (First and Middle):

\_\_\_\_\_

Family (Last) Name:

\_\_\_\_\_

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Date

**Electronic Acknowledgement Receipt**

<b>EFS ID:</b>	15455240
<b>Application Number:</b>	13849606
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	6059
<b>Title of Invention:</b>	Processing with Compact Arithmetic Processing Element
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates
<b>Customer Number:</b>	24208
<b>Filer:</b>	Robert Plotkin/Karen Del Greco
<b>Filer Authorized By:</b>	Robert Plotkin
<b>Attorney Docket Number:</b>	A0006-1001C2
<b>Receipt Date:</b>	08-APR-2013
<b>Filing Date:</b>	
<b>Time Stamp:</b>	12:34:15
<b>Application Type:</b>	Utility under 35 USC 111(a)

**Payment information:**

Submitted with Payment	no
------------------------	----

**File Listing:**

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Oath or Declaration filed	Declaration_A0006-1001C2.pdf	827679 faffbe57bf348a89c9e80f04ada2a23980ecc07b	no	4

**Warnings:****Information:**

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

**New Applications Under 35 U.S.C. 111**

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

**National Stage of an International Application under 35 U.S.C. 371**

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

**New International Application Filed with the USPTO as a Receiving Office**

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

## Power of Attorney

*Equivalent to USPTO Form PTO/AIA/82B*

### 1. Revocation of Previous Powers of Attorney

I/we hereby revoke all previous powers of attorney given in the application identified in the attached transmittal letter (form PTO/AIA/82A or equivalent).

### 2. Appointment of Practitioner(s)

I/we hereby appoint the Practitioner(s) associated with Customer Number 24208 (which is associated with the law firm of Robert Plotkin, P.C.) as my/our attorney(s)/agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A or equivalent).

### 3. Change of Correspondence Address

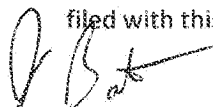
Please recognize or change the correspondence address for the application identified in the attached transmittal letter to the address associated with the above-mentioned customer number 24208 (which is associated with the law firm of Robert Plotkin, P.C.).



**4. Signature(s)****a. Applicant 1**

I am an Applicant:

- ☐ Inventor or Joint Inventor  
☐ Legal Representative of a Deceased or Legally Incapacitated Inventor  
☒ Assignee or Person to Whom the Inventor is Under an Obligation to Assign  
☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document)

  
 Signature

29 March 2013  
 Date

JOSEPH BATES  
 Name

Telephone

PRESIDENT  
 Title

Singular Computing LLC  
 Company

**b. Applicant 2**

I am an Applicant:

- ☐ Inventor or Joint Inventor  
☐ Legal Representative of a Deceased or Legally Incapacitated Inventor  
☐ Assignee or Person to Whom the Inventor is Under an Obligation to Assign  
☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document)

Signature

Date

Name

Telephone

Title

Company

c. Applicant 3

I am an Applicant:

- ☐ Inventor or Joint Inventor
- ☐ Legal Representative of a Deceased or Legally Incapacitated Inventor
- ☐ Assignee or Person to Whom the Inventor is Under an Obligation to Assign
- ☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document)

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Date

\_\_\_\_\_  
Name

\_\_\_\_\_  
Telephone

\_\_\_\_\_  
Title

\_\_\_\_\_  
Company

d. Applicant 4

I am an Applicant:

- ☐ Inventor or Joint Inventor
- ☐ Legal Representative of a Deceased or Legally Incapacitated Inventor
- ☐ Assignee or Person to Whom the Inventor is Under an Obligation to Assign
- ☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document)

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Date

\_\_\_\_\_  
Name

\_\_\_\_\_  
Telephone

\_\_\_\_\_  
Title

\_\_\_\_\_  
Company

e. Applicant 5

I am an Applicant:

- ☐ Inventor or Joint Inventor
- ☐ Legal Representative of a Deceased or Legally Incapacitated Inventor
- ☐ Assignee or Person to Whom the Inventor is Under an Obligation to Assign
- ☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document)

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Date

\_\_\_\_\_  
Name

\_\_\_\_\_  
Telephone

\_\_\_\_\_  
Title

\_\_\_\_\_  
Company

f. Applicant 6

I am an Applicant:

- ☐ Inventor or Joint Inventor
- ☐ Legal Representative of a Deceased or Legally Incapacitated Inventor
- ☐ Assignee or Person to Whom the Inventor is Under an Obligation to Assign
- ☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document)

\_\_\_\_\_  
Signature

\_\_\_\_\_  
Date

\_\_\_\_\_  
Name

\_\_\_\_\_  
Telephone

\_\_\_\_\_  
Title

\_\_\_\_\_  
Company

**Electronic Acknowledgement Receipt**

<b>EFS ID:</b>	15455305
<b>Application Number:</b>	13849606
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	6059
<b>Title of Invention:</b>	Processing with Compact Arithmetic Processing Element
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates
<b>Customer Number:</b>	24208
<b>Filer:</b>	Robert Plotkin/Karen Del Greco
<b>Filer Authorized By:</b>	Robert Plotkin
<b>Attorney Docket Number:</b>	A0006-1001C2
<b>Receipt Date:</b>	08-APR-2013
<b>Filing Date:</b>	
<b>Time Stamp:</b>	12:38:24
<b>Application Type:</b>	Utility under 35 USC 111(a)

**Payment information:**

Submitted with Payment	no
------------------------	----

**File Listing:**

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	POATransmittal_A0006-1001C2.pdf	93227 c1bad6aa59d885976abbba278335254ebfc76fba	no	1

**Warnings:****Information:**

2	Oath or Declaration filed	POA_A0006-1001C2.pdf	1036556 066279391ef5f6ccaa8f09fb53db7a6426927da7	no	4
<b>Warnings:</b>					
<b>Information:</b>					
<b>Total Files Size (in bytes):</b>				1129783	
<p><b>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</b></p> <p><b><u>New Applications Under 35 U.S.C. 111</u></b>  <b>If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</b></p> <p><b><u>National Stage of an International Application under 35 U.S.C. 371</u></b>  <b>If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</b></p> <p><b><u>New International Application Filed with the USPTO as a Receiving Office</u></b>  <b>If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</b></p>					



## Transmittal for Power of Attorney to One or More Registered Practitioners

*Equivalent to USPTO Form PTO/AIA/82A*

Application Number: 13/849,606 Attorney Docket Number: A0006-1001C2  
Filing Date: 03/25/2013 Examiner Name: Michael D. Yaary  
First Named Inventor: BATES, Joseph Art Unit: N/A  
Title of Invention: Processing with Compact Arithmetic Processing Element

This form is being submitted with the attached Power of Attorney by Applicant (PTO/AIA/82B or equivalent) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5.

Signature of Patent Practitioner

<u>/Robert Plotkin, Reg#43861/</u>	<u>April 8, 2013</u>
Signature	Date
<u>Robert Plotkin</u>	<u>978-318-9914</u>
Name	Telephone

43,861

Registration Number

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

<b>Application Data Sheet 37 CFR 1.76</b>		Attorney Docket Number	A0006-1001C2
		Application Number	
Title of Invention	Processing with Compact Arithmetic Processing Element		
<p>The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.</p> <p>This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.</p>			

**Secrecy Order 37 CFR 5.2**

<input type="checkbox"/> Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)
--

**Inventor Information:**

Inventor 1 <span style="float: right;">Remove</span>				
Legal Name				
Prefix	Given Name	Middle Name	Family Name	Suffix
	Joseph		Bates	
Residence Information (Select One) <input checked="" type="radio"/> US Residency <input type="radio"/> Non US Residency <input type="radio"/> Active US Military Service				
City	Newton	State/Province	MA	Country of Residence i US
Mailing Address of Inventor:				
Address 1		Singular Computing LLC		
Address 2		10 Regent Street		
City	Newton	State/Province	MA	
Postal Code	02465	Country i	US	
All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the Add button. <span style="float: right;">Add</span>				

**Correspondence Information:**

Enter either Customer Number or complete the Correspondence Information section below. For further information see 37 CFR 1.33(a).			
<input type="checkbox"/> An Address is being provided for the correspondence Information of this application.			
Customer Number	24208		
Email Address	rplotkin@rplotkin.com	Add Email	Remove Email

**Application Information:**

Title of the Invention	Processing with Compact Arithmetic Processing Element		
Attorney Docket Number	A0006-1001C2	Small Entity Status Claimed	<input checked="" type="checkbox"/>
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	11	Suggested Figure for Publication (if any)	1

<b>Application Data Sheet 37 CFR 1.76</b>		Attorney Docket Number	A0006-1001C2
		Application Number	
Title of Invention	Processing with Compact Arithmetic Processing Element		

**Publication Information:**

<input type="checkbox"/>	Request Early Publication (Fee required at time of Request 37 CFR 1.219)
<input type="checkbox"/>	<b>Request Not to Publish.</b> I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application <b>has not and will not</b> be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

**Representative Information:**

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.			
Please Select One:			
<input checked="" type="radio"/>	Customer Number	<input type="radio"/>	US Patent Practitioner
<input type="radio"/>	Limited Recognition (37 CFR 11.9)		
Customer Number	24208		

**Domestic Benefit/National Stage Information:**

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.					
Prior Application Status	Pending		<a href="#">Remove</a>		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)		
	Continuation of	13399884	2012-02-17		
Prior Application Status	Patented		<a href="#">Remove</a>		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
13399884	Continuation of	12816201	2010-06-15	8150902	2012-04-03
Prior Application Status	Expired		<a href="#">Remove</a>		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)		
12816201	non provisional of	61218691	2009-06-19		
Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the <b>Add</b> button.					<a href="#">Add</a>

**Foreign Priority Information:**

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

<b>Application Data Sheet 37 CFR 1.76</b>		Attorney Docket Number	A0006-1001C2
		Application Number	
Title of Invention	Processing with Compact Arithmetic Processing Element		

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(d). When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX) the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(h)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

<b>Remove</b>			
Application Number	Country <sup>i</sup>	Filing Date (YYYY-MM-DD)	Access Code <sup>j</sup> (if applicable)
Additional Foreign Priority Data may be generated within this form by selecting the <b>Add</b> button.			<b>Add</b>

## Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

<input type="checkbox"/>	This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
--------------------------	--

## Authorization to Permit Access:

<input checked="" type="checkbox"/>	<b>Authorization to Permit Access to the Instant Application by the Participating Offices</b>
<p>If checked, the undersigned hereby grants the USPTO authority to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the World Intellectual Property Office (WIPO), and any other intellectual property offices in which a foreign application claiming priority to the instant patent application is filed access to the instant patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, KIPO, WIPO, or other intellectual property office in which a foreign application claiming priority to the instant patent application is filed to have access to the instant patent application.</p> <p>In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the instant patent application with respect to: 1) the instant patent application-as-filed; 2) any foreign application to which the instant patent application claims priority under 35 U.S.C. 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the instant patent application; and 3) any U.S. application-as-filed from which benefit is sought in the instant patent application.</p> <p>In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date of filing this Authorization.</p>	

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

<b>Application Data Sheet 37 CFR 1.76</b>	Attorney Docket Number	A0006-1001C2
	Application Number	
Title of Invention	Processing with Compact Arithmetic Processing Element	

## Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.			
<b>Applicant 1</b>			<a href="#">Remove</a>
If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.			
<a href="#">Clear</a>			
<input checked="" type="radio"/> Assignee	<input type="radio"/> Legal Representative under 35 U.S.C. 117	<input type="radio"/> Joint Inventor	
<input type="radio"/> Person to whom the inventor is obligated to assign.	<input type="radio"/> Person who shows sufficient proprietary interest		
If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:			
Name of the Deceased or Legally Incapacitated Inventor :			
If the Applicant is an Organization check here. <input checked="" type="checkbox"/>			
Organization Name	Singular Computing LLC		
<b>Mailing Address Information:</b>			
Address 1	10 Regent Street		
Address 2			
City	Newton	State/Province	MA
Country	US	Postal Code	02465
Phone Number		Fax Number	
Email Address			
Additional Applicant Data may be generated within this form by selecting the Add button. <a href="#">Add</a>			

## Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.
---



Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

<b>Application Data Sheet 37 CFR 1.76</b>		Attorney Docket Number	A0006-1001C2
		Application Number	
Title of Invention	Processing with Compact Arithmetic Processing Element		

<b>Assignee 1</b>				
Complete this section only if non-applicant assignee information is desired to be included on the patent application publication in accordance with 37 CFR 1.215(b). Do not include in this section an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest), as the patent application publication will include the name of the applicant(s).				
				<a href="#">Remove</a>
If the Assignee is an Organization check here. <input type="checkbox"/>				
Prefix	Given Name	Middle Name	Family Name	Suffix
<b>Mailing Address Information:</b>				
Address 1				
Address 2				
City		State/Province		
Country i		Postal Code		
Phone Number		Fax Number		
Email Address				
Additional Assignee Data may be generated within this form by selecting the Add button.				<a href="#">Add</a>

**Signature:**[Remove](#)

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications					
<b>Signature</b>	/Robert Plotkin, Reg#43861/			<b>Date (YYYY-MM-DD)</b>	2013-03-25
<b>First Name</b>	Robert	<b>Last Name</b>	Plotkin	<b>Registration Number</b>	43861
Additional Signature may be generated within this form by selecting the Add button.					<a href="#">Add</a>

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

## Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Attorney Docket No. A0006-1001C2

**APPLICATION FOR  
UNITED STATES LETTERS PATENT**

**Title:       Processing with Compact Arithmetic  
              Processing Element**

**Inventor(s):   Joseph Bates**

## **Processing with Compact Arithmetic Processing Element**

### **Cross-Reference to Related Applications**

**[0001]** This application is a continuation of U.S. Patent Application Serial Number 13/399,884, filed on February 17, 2012, entitled, “Processing with Compact Arithmetic Processing Element,” now U.S. Patent Number 8,407,273 (Attorney Docket No. A0006-1001C1); which is a continuation of U.S. Patent Application Serial Number 12/816,201, filed on June 15, 2010, entitled, “Processing with Compact Arithmetic Processing Element,” now U.S. Patent Number 8,150,902 (Attorney Docket No. A0006-1001); which claims the benefit of U.S. Provisional Patent Application Serial Number 61/218,691, filed on June 19, 2009, entitled, “Massively Parallel Processing with Compact Arithmetic Element” (Attorney Docket No. A0006-1001L); all of which are hereby incorporated by reference herein.

### **Copyright Notice**

**[0002]** A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

### **Background**

**[0003]** The ability to compute rapidly has become enormously important to humanity. Weather and climate prediction, medical applications (such as drug design and non-invasive imaging), national defense, geological exploration, financial modeling, Internet search, network communications, scientific research in varied fields, and even the design of new computing hardware have each become dependent on the ability to rapidly perform massive amounts of calculation. Future progress, such as the computer-aided design of complex nano-scale systems or development of consumer products that can see,

Attorney Docket No. A0006-1001C2

hear, and understand, will demand economical delivery of even greater computing power.

**[0004]** Gordon Moore's prediction, that computing performance per dollar would double every two years, has proved valid for over 30 years and looks likely to continue in some form. But despite this rapid exponential improvement, the reality is that the inherent computing power available from silicon has grown far more quickly than it has been made available to software. In other words, although the theoretical computing power of computing hardware has grown exponentially, the interfaces through which software is required to access the hardware limits the ability of software to use hardware to perform computations at anything approaching the hardware's theoretical maximum computing power.

**[0005]** Consider a modern silicon microprocessor chip containing about one billion transistors, clocked at roughly 1 GHz. On each cycle the chip delivers approximately one useful arithmetic operation to the software it is running. For instance, a value might be transferred between registers, another value might be incremented, perhaps a multiply is accomplished. This is not terribly different from what chips did 30 years ago, though the clock rates are perhaps a thousand times faster today.

**[0006]** Real computers are built as physical devices, and the underlying physics from which the machines are built often exhibits complex and interesting behavior. For example, a silicon MOSFET transistor is a device capable of performing interesting non-linear operations, such as exponentiation. The junction of two wires can add currents. If configured properly, a billion transistors and wires should be able to perform some significant fraction of a billion interesting computational operations within a few propagation delays of the basic components (a "cycle" if the overall design is a traditional digital design). Yet, today's CPU chips use their billion transistors to enable software to perform merely a few such operations per cycle, not the significant fraction of the billion that might be possible.



### **Summary**

**[0007]** Embodiments of the present invention are directed to a processor or other device, such as a programmable and/or massively parallel processor or other device, which includes processing elements designed to perform arithmetic operations (possibly but not necessarily including, for example, one or more of addition, multiplication, subtraction, and division) on numerical values of low precision but high dynamic range ("LPHDR arithmetic"). Such a processor or other device may, for example, be implemented on a single chip. Whether or not implemented on a single chip, the number of LPHDR arithmetic elements in the processor or other device in certain embodiments of the present invention significantly exceeds (e.g., by at least 20 more than three times) the number of arithmetic elements in the processor or other device which are designed to perform high dynamic range arithmetic of traditional precision (such as 32 bit or 64 bit floating point arithmetic).

**[0008]** In some embodiments, "low precision" processing elements perform arithmetic operations which produce results that frequently differ from exact results by at least .1% (one tenth of one percent). This is far worse precision than the widely used IEEE 754 single precision floating point standard. Programmable embodiments of the present invention may be programmed with algorithms that function adequately despite these unusually large relative errors. In some embodiments, the processing elements have "high dynamic range" in the sense that they are capable of operating on inputs and/or producing outputs spanning a range at least as large as from one millionth to one million.

### **Brief Description of the Drawings**

**[0009]** FIG. 1 is an example overall design of a SIMD processor according to one embodiment of the present invention.

**[0010]** FIG. 2 is an example of the Processing Element Array of a SIMD processor according to one embodiment of the present invention.

**[0011]** FIG. 3 is an example of how a Processing Element in a Processing Element Array communicates data with other parts of the processor

Attorney Docket No. A0006-1001C2

according to one embodiment of the present invention.

**[0012]** FIG. 4 is an example design for a Processing Element according to one embodiment of the present invention.

**[0013]** FIG. 5 is an example LPHDR data word format according to one embodiment of the present invention.

**[0014]** FIG. 6 is an example design for an LPHDR arithmetic unit according to one embodiment of the present invention.

**[0015]** FIG. 7 is an original image.

**[0016]** FIG. 8 is an image blurred by a blur kernel according to one embodiment of the present invention.

**[0017]** FIG. 9 is an image produced by Richardson Lucy deconvolution using floating point arithmetic according to one embodiment of the present invention.

**[0018]** FIG. 10 is an image produced by Richardson Lucy deconvolution using LPHDR floating point arithmetic with added noise (fp+noise) according to one embodiment of the present invention.

**[0019]** FIG. 11 is an image produced by Richardson Lucy deconvolution using LPHDR logarithmic arithmetic (lns) according to one embodiment of the present invention.

### **Detailed Description**

**[0020]** As described above, today's CPU chips make inefficient use of their transistors. For example, a conventional CPU chip containing a billion transistors might enable software to perform merely a few operations per clock cycle. Although this is highly inefficient, those having ordinary skill in the art design CPUs in this way for what are widely accepted to be valid reasons. For example, such designs satisfy the (often essential) requirement for software compatibility with earlier designs. Furthermore, they deliver great precision, performing exact arithmetic with integers typically 32 or 64 bits long and performing rather accurate and widely standardized arithmetic with 32 and 64 bit floating point numbers. Many applications need this kind of precision. As a

result, conventional CPUs typically are designed to provide such precision, using on the order of a million transistors to implement the arithmetic operations.

**[0021]** There are many economically important applications, however, which are not especially sensitive to precision and that would greatly benefit, in the form of application performance per transistor, from the ability to draw upon a far greater fraction of the computing power inherent in those million transistors. Current architectures for general purpose computing fail to deliver this power.

**[0022]** Because of the weaknesses of conventional computers, such as typical microprocessors, other kinds of computers have been developed to attain higher performance. These machines include single instruction stream/multiple data stream (SIMD) designs, multiple instruction stream/multiple data stream (MIMD) designs, reconfigurable architectures such as field programmable gate arrays (FPGAs), and graphics processing unit designs (GPUs) which, when applied to general purpose computing, may be viewed as single instruction stream/multiple thread (SIMT) designs.

**[0023]** SIMD machines follow a sequential program, with each instruction performing operations on a collection of data. They come in two main varieties: vector processors and array processors. Vector processors stream data through a processing element (or small collection of such elements). Each component of the data stream is processed similarly. Vector machines gain speed by eliminating many instruction fetch/decode operations and by pipelining the processor so that the clock speed of the operations is increased.

**[0024]** Array processors distribute data across a grid of processing elements (PEs). Each element has its own memory. Instructions are broadcast to the PEs from a central control unit, sequentially. Each PE performs the broadcast instruction on its local data (often with the option to sit idle that cycle). Array processors gain speed by using silicon efficiently—using just one instruction fetch/decode unit to drive many small simple execution units in parallel.

**[0025]** Array processors have been built using fixed point arithmetic at a wide variety of bit widths, such as 1, 4, 8, and wider, and using floating point

Attorney Docket No. A0006-1001C2

arithmetic. Small bit widths allow the processing elements to be small, which allows more of them to fit in the computer, but many operations must be carried out in sequence to perform conventional arithmetic calculations. Wider widths allow conventional arithmetic operations to be completed in a single cycle. In practice, wider widths are desirable. Machines that were originally designed with small bit widths, such as the Connection Machine-1 and the Goodyear Massively Parallel Processor, which each used 1 bit wide processing elements, evolved toward wider data paths to better support fast arithmetic, producing machines such as the Connection Machine-2 which included 32 bit floating point hardware and the MasPar machines which succeeded the Goodyear machine and provided 4 bit processing elements in the MasPar-1 and 32 bit processing elements in the MasPar-2.

**[0026]** Array processors also have been designed to use analog representations of numbers and analog circuits to perform computations. The SCAMP is such a machine. These machines provide low precision arithmetic, in which each operation might introduce perhaps an error of a few percentage points in its results. They also introduce noise into their computations, so the computations are not repeatable. Further, they represent only a small range of values, corresponding for instance to 8 bit fixed point values rather than providing the large dynamic range of typical 32 or 64 bit floating point representations. Given these limitations, the SCAMP was not intended as a general purpose computer, but instead was designed and used for image processing and for modeling biological early vision processes. Such applications do not require a full range of arithmetic operations in hardware, and the SCAMP, for example, omits general division and multiplication from its design.

**[0027]** While SIMD machines were popular in the 1980s, as price/performance for microprocessors improved designers began building machines from large collections of communicating microprocessors. These MIMD machines are fast and can have price/performance comparable to their component microprocessors, but they exhibit the same inefficiency as those components in that they deliver to their software relatively little computation per

transistor.

**[0028]** Field Programmable Gate Arrays (FPGAs) are integrated circuits containing a large grid of general purpose digital elements with reconfigurable wiring between those elements. The elements originally were single digital gates, such as AND and OR gates, but evolved to larger elements that could, for instance, be programmed to map 6 inputs to 1 output according to any Boolean function. This architecture allows the FPGA to be configured from external sources to perform a wide variety of digital computations, which allows the device to be used as a co-processor to a CPU to accelerate computation. However, arithmetic operations such as multiplication and division on integers, and especially on floating point numbers, require many gates and can absorb a large fraction of an FPGA's general purpose resources. For this reason, modern FPGAs often devote a significant portion of their area to providing dozens or hundreds of multiplier blocks, which can be used instead of general purpose resources for computations requiring multiplication. These multiplier blocks typically perform 18 bit or wider integer multiplies, and use many transistors, as similar multiplier circuits do when they are part of a general purpose CPU.

**[0029]** Existing Field Programmable Analog Arrays (FPAAs) are analogous to FPGAs, but their configurable elements perform analog processing. These devices generally are intended to do signal processing, such as helping model neural circuitry. They are relatively low precision, have relatively low dynamic range, and introduce noise into computation. They have not been designed as, or intended for use as, general purpose computers. For instance, they are not seen by those having ordinary skill in the art as machines that can run the variety of complex algorithms with floating point arithmetic that typically run on high performance digital computers.

**[0030]** Finally, Graphics Processing Units (GPUs) are a variety of parallel processor that evolved to provide high speed graphics capabilities to personal computers. They offer standard floating point computing abilities with very high performance for certain tasks. Their computing model is sometimes based on having thousands of nearly identical threads of computing (SIMT),



Attorney Docket No. A0006-1001C2

which are executed by a collection of SIMD-like internal computing engines, each of which is directed and redirected to perform work for which a slow external DRAM memory has provided data. Like other machines that implement standard floating point arithmetic, they use many transistors for that arithmetic. They are as wasteful of those transistors, in the sense discussed above, as are general purpose CPUs.

**[0031]** Some GPUs include support for 16 bit floating point values (sometimes called the “Half” format). The GPU manufacturers, currently such as NVIDIA or AMD/ATI, describe this capability as being useful for rendering images with higher dynamic range than the usual 32 bit RGBA format, which uses 8 bits of fixed point data per color, while also saving space over using 32 bit floating point for color components. The special effects movie firm Industrial Light and Magic (ILM) independently defined an identical representation in their OpenEXR standard, which they describe as “a high dynamic-range (HDR) image file format developed by Industrial Light & Magic for use in computer imaging applications.” Wikipedia (late 2008) describes the 16 bit floating point representation thusly: “This format is used in several computer graphics environments including OpenEXR, OpenGL, and D3DX. The advantage over 8-bit or 16-bit binary integers is that the increased dynamic range allows for more detail to be preserved in highlights and shadows. The advantage over 32-bit single precision binary formats is that it requires half the storage and bandwidth.”

**[0032]** When a graphics processor includes support for 16 bit floating point, that support is alongside support for 32 bit floating point, and increasingly, 64 bit floating point. That is, the 16 bit floating point format is supported for those applications that want it, but the higher precision formats also are supported because they are believed to be needed for traditional graphics applications and also for so called “general purpose” GPU applications. Thus, existing GPUs devote substantial resources to 32 (and increasingly 64) bit arithmetic and are wasteful of transistors in the sense discussed above.

**[0033]** The variety of architectures mentioned above are all attempts to get more performance from silicon than is available in a traditional processor



design. But designers of traditional processors also have been struggling to use the enormous increase in available transistors to improve performance of their machines. These machines often are required, because of history and economics, to support large existing instruction sets, such as the Intel x86 instruction set. This is difficult, because of the law of diminishing returns, which does not enable twice the performance to be delivered by twice the transistor count. One facet of these designers' struggle has been to increase the precision of arithmetic operations, since transistors are abundant and some applications could be sped up significantly if the processor natively supported long (e.g., 64 bit) numbers. With the increase of native fixed point precision from 8 to 16 to 32 to 64 bits, and of floating point from 32 to 64 and sometimes 128 bits, programmers have come to think in terms of high precision and to develop algorithms based on the assumption that computer processors provide such precision, since it comes as an integral part of each new generation of silicon chips and thus is "free."

**[0034]** Embodiments of the present invention efficiently provide computing power using a fundamentally different approach than those described above. In particular, embodiments of the present invention are directed to computer processors or other devices which use low precision high dynamic range (LPHDR) processing elements to perform computations (such as arithmetic operations).

**[0035]** One variety of LPHDR arithmetic represents values from one millionth up to one million with a precision of about 0.1%. If these values were represented and manipulated using the methods of floating point arithmetic, they would have binary mantissas of no more than 10 bits plus a sign bit and binary exponents of at least 5 bits plus a sign bit. However, the circuits to multiply and divide these floating point values would be relatively large. One example of an alternative embodiment is to use a logarithmic representation of the values. In such an approach, the values require the same number of bits to represent, but multiplication and division are implemented as addition and subtraction, respectively, of the logarithmic representations. Addition and subtraction may be

Attorney Docket No. A0006-1001C2

implemented efficiently as described below. As a result, the area of the arithmetic circuits remains relatively small and a greater number of computing elements can be fit into a given area of silicon. This means the machine can perform a greater number of operations per unit of time or per unit power, which gives it an advantage for those computations able to be expressed in the LPHDR framework.

**[0036]** Another embodiment is to use analog representations and processing mechanisms. Analog implementation of LPHDR arithmetic has the potential to be superior to digital implementation, because it tends to use the natural analog physics of transistors or other physical devices instead of using only the digital subset of the device's behavior. This fuller use of the devices' natural abilities may permit smaller mechanisms for doing LPHDR arithmetic. In recent years, in the field of silicon circuitry, analog methods have been supplanted by digital methods. In part, this is because of the ease of doing digital design compared to analog design. Also in part, it is because of the continued rapid scaling of digital technology ("Moore's Law") compared to analog technology. In particular, at deep submicron dimensions, analog transistors no longer work as they had in prior generations of larger-scale technology. This change of familiar behavior has made analog design still harder in recent years. However, digital transistors are in fact analog transistors used in a digital way, meaning digital circuits are really analog circuits designed to attempt to switch the transistors between completely on and completely off states. As scaling continues, even this use of transistors is starting to come face to face with the realities of analog behavior. Scaling of transistors for digital use is expected either to stall or to require digital designers increasingly to acknowledge and work with analog issues. For these reasons, digital embodiments may no longer be easy, reliable, and scalable, and analog embodiments of LPHDR arithmetic may come to dominate commercial architectures.

**[0037]** Because LPHDR processing elements are relatively small, a single processor or other device may include a very large number of LPHDR processing elements, adapted to operate in parallel with each other, and

therefore may constitute a massively parallel LPHDR processor or other device. Such a processor or other device has not been described or practiced as a means of doing general purpose computing by those having ordinary skill in the art for at least two reasons. First, it is commonly believed by those having ordinary skill in the art, that LPHDR computation, and in particular massive amounts of LPHDR computation, whether performed in a massively parallel way or not, is not practical as a substrate for moderately general computing. Second, it is commonly believed by those having ordinary skill in the art that massive amounts of even high precision computation on a single chip or in a single machine, as is enabled by a compact arithmetic processing unit, is not useful without a corresponding increase in bandwidth between processing elements within the machine and into and out of the machine because computing is wire limited and arithmetic can be considered to be available at no cost.

**[0038]** Despite these views—that massive amounts of arithmetic on a chip or in a massively parallel machine are not useful, and that massive amounts of LPHDR arithmetic are even worse—embodiments of the present invention disclosed herein demonstrate that massively parallel LPHDR designs are in fact useful and provide significant practical benefits in at least several significant applications.

**[0039]** To conclude, modern digital computing systems provide high precision arithmetic, but that precision is costly. A modern double precision floating point multiplier may require on the order of a million transistors, even though only a handful of transistors is required to perform a low precision multiplication. Despite the common belief among those having ordinary skill in the art that modern applications require high precision processing, in fact a variety of useful algorithms function adequately at much lower precision. As a result, such algorithms may be performed by processors or other devices implemented according to embodiments of the present invention, which come closer to achieving the goal of using a few transistors to multiply and a wire junction to add, thus enabling massively parallel arithmetic computation to be performed with relatively small amounts of physical resources (such as a single

Attorney Docket No. A0006-1001C2

silicon chip). Although certain specialized tasks can function at low precision, it is not obvious, and in fact has been viewed as clearly false by those having ordinary skill in the art, that relatively general purpose computing such as is typically performed today on general purpose computers can be done at low precision. However, in fact a variety of useful and important algorithms can be made to function adequately at much lower than 32 bit precision in a massively parallel computing framework, and certain embodiments of the present invention support such algorithms, thereby offering much more efficient use of transistors, and thereby provide improved speed, power, and/or cost, compared to conventional computers.

**[0040]** Various computing devices implemented according to embodiments of the present invention will now be described. Some of these embodiments may be an instance of a SIMD computer architecture. Other architectures may be used, such as MIMD architectures, programmable array architectures (such as FPGAs and FPAAs), or GPU/SIMT architectures. The techniques disclosed herein may, for example, be implemented using any processor or other device having such an existing architecture, and replacing or augmenting some or all existing arithmetic units in the processor or other device, if any, with LPHDR arithmetic units in any of the ways disclosed herein. Devices implemented according to embodiments of the present invention, however, need not start with an existing processor design, but instead may be designed from scratch to include LPHDR arithmetic units within any of the architectures just described, or any other architecture.

**[0041]** Embodiments of the present invention may, for example, be implemented using the architecture of a particular kind of SIMD computer, the array processor. There are many variations and specific instances of array processors described in the scientific and commercial literature. Examples include the Illiac 4, the Connection Machine 1 and 2, the Goodyear MPP, and the MasPar line of computers.

**[0042]** Embodiments of the present invention need not, however, be implemented as SIMD computers. For example, embodiments of the present

Attorney Docket No. A0006-1001C2

invention may be implemented as FPGAs, FPAAAs, or related architectures that provide for flexible connectivity of a set of processing elements. For example, embodiments of the present invention may be implemented as GPU/SIMTs and as MIMDs, among others. For example, embodiments of the present invention may be implemented as any kind of machine which uses LPHDR arithmetic processing elements to provide computing using a small amount of resources (e.g., transistors or volume) compared with traditional architectures.

Furthermore, references herein to "processing elements" within embodiments of the present invention should be understood more generally as any kind of execution unit, whether for performing LPHDR operations or otherwise.

**[0043]** An example SIMD computing system 100 is illustrated in FIG. 1. The computing system 100 includes a collection of many processing elements (PEs) 104. Sometimes present are a control unit (CU) 106, an I/O unit (IOU) 108, various Peripheral devices 110, and a Host computer 102. The collection of PEs is referred to herein as "the Processing Element Array" (PEA), even though it need not be two-dimensional or an array or grid or other particular layout. Some machines include additional components, such as an additional memory system called the "Staging Memory" in the Goodyear MPP, but these additional elements are neither essential in the computer nor needed to understand embodiments of the present invention and therefore are omitted here for clarity of explanation. One embodiment of the present invention is a SIMD computing system of the kind shown in FIG. 1, in which one or more (e.g., all) of the PEs in the PEA 104 are LPHDR elements, as that term is used herein.

**[0044]** The Host 102 is responsible for overall control of the computing system 100. It performs the serial, or mostly serial, computation typical of a traditional uni-processor. The Host 102 could have more complicated structure, of course, including parallelism of various sorts. Indeed a heterogeneous computing system combining multiple computing architectures in a single machine is a good use for embodiments of the present invention.

**[0045]** A goal of the Host 102 is to have the PEA 104 perform massive amounts of computation in a useful way. It does this by causing the PEs to



Attorney Docket No. A0006-1001C2

perform computations, typically on data stored locally in each PE, in parallel with one another. If there are many PEs, much work gets done during each unit of time.

**[0046]** The PEs in the PEA 104 may be able to perform their individual computations roughly as fast as the Host 102 performs its computations. This means it may be inefficient to have the Host 102 attempt to control the PEA 104 on a time scale as fine as the Host's or PEA's minimal time step. (This minimal time, in a traditional digital design, would be the clock period.) For this reason, the specialized control unit (CU) 106 may be included in the architecture. The CU 106 has the primary task of retrieving and decoding instructions from an instruction memory, which conceptually is part of the CU 106, and issuing the partially decoded instructions to all the PEs in the PEA 104. (This may be viewed by the CU software as happening roughly simultaneously for all the PEs, though it need not literally be synchronous, and in fact it may be effective to use an asynchronous design in which multiple instructions at different stages of completion simultaneously propagate gradually across the PEA, for instance as a series of wave fronts.)

**[0047]** In a design which includes the CU 106, the Host 102 typically will load the instructions (the program) for the PEA 104 into the CU instruction memory (not shown in FIG. 1), then instruct the CU 106 to interpret the program and cause the PEA 104 to compute according to the instructions. The program may, for example, look generally similar to a typical machine language program, with instructions to cause data movement, logical operations, arithmetic operations, etc., in and between the PEs and other instructions to do similar operations together with control flow operations within the CU 106. Thus, the CU 106 may run a typical sort of program, but with the ability to issue massively parallel instructions to the PEA 104.

**[0048]** In order to get data into and out of the CU 106 and PEA 104, the I/O Unit 108 may interface the CU 106 and PEA 104 with the Host 102, the Host's memory (not shown in FIG. 1), and the system's Peripherals 110, such as external storage (e.g., disk drives), display devices for visualization of the



Attorney Docket No. A0006-1001C2

computational results, and sometimes special high bandwidth input devices (e.g., vision sensors). The PEA's ability to process data far faster than the Host 102 makes it useful for the IOU 108 to be able to completely bypass the Host 102 for some of its data transfers. Also, the Host 102 may have its own ways of communicating with the Peripherals 110.

**[0049]** The particular embodiment illustrated in FIG. 1 is shown merely for purposes of example and does not constitute a limitation of the present invention. For example, alternatively the functions performed by the CU 106 could instead be performed by the Host 102 with the CU 106 omitted. The CU 106 could be implemented as hardware distant from the PEA 104 (e.g., off-chip), or the CU 106 could be near to the PEA 104 (e.g., on-chip). I/O could be routed through the CU 106 with the IOU 108 omitted or through the separate I/O controller 108, as shown. Furthermore, the Host 102 is optional; the CU 106 may include, for example, a CPU, or otherwise include components sufficient to replace the functions performed by the Host 102. The Peripherals 110 shown in FIG. 1 are optional. The design shown in FIG. 1 could have a special memory, such as the Goodyear MPP's "staging memory," which provides an intermediate level of local storage. Such memory could, for example, be bonded to the LPHDR chip using 3D fabrication technology to provide relatively fast parallel access to the memory from the PEs in the PEA 104.

**[0050]** The PEA 104 itself, besides communicating with the CU 106 and IOU 108 and possibly other mechanisms, has ways for data to move within the array. For example, the PEA 104 may be implemented such that data may move from PEs only to their nearest neighbors, that is, there are no long distance transfers. FIGS. 2 and 3 show embodiments of the present invention which use this approach, where the nearest neighbors are the four adjacent PEs toward the North, East, West, and South, called a NEWS design. For example, FIG. 2 shows a subset of the PEs in PEA 104, namely PE 202, PE 204, PE 206, PE 208, and PE 210. When the CU 106 issues data movement instructions, all the PEs access data from or send data to their respective specified nearest neighbor. For instance, every PE might access a specified data value in its

neighbor to the West and copy it into its own local storage. In some embodiments, such as some analog embodiments, these kinds of transfers may result in some degradation of the value copied.

**[0051]** FIG. 3 shows a PE 302 that includes data connections to the IOU 108. PE 302 is connected at the North to PE 304, at the East to PE 306, at the South to PE 308, and at the West to PE 310. However, driving signals from inside the PEA 104 out to the IOU 108 usually requires a physically relatively large driving circuit or analogous mechanism. Having those at every PE may absorb much of the available resources of the hardware implementation technology (such as VLSI area). In addition, having independent connections from every PE to the IOU 108 means many such connections, and long connections, which also may absorb much of the available hardware resources. For these reasons, the connections between the PEs and the IOU 108 may be limited to those PEs at the edges of the PE array 104. In this case, to get data out of, and perhaps into, the PEA 104, the data is read and written at the edges of the array and CU instructions are performed to shift data between the edges and interior of the PEA 104. The design may permit data to be pushed from the IOU 108 inward to any PE in the array using direct connections, but may require readout to occur by using the CU 106 to shift data to the edges where it can be read by the IOU 108.

**[0052]** Connections between the CU 106 and PEA 104 have analogous variations. One design may include the ability to drive instructions into all the PEs roughly simultaneously, but another approach is to have the instructions flow gradually (for instance, shift in discrete time steps) across the PEA 104 to reach the PEs. Some SIMD designs, which may be implemented in embodiments of the present invention, have a facility by which a "wired-or" or "wired-and" of the state of every PE in the PEA 104 can be read by the CU 106 in approximately one instruction delay time.

**[0053]** There are many well studied variations on these matters in the literature, any of which may be incorporated into embodiments of the present invention. For example, an interconnect, such as an 8-way local interconnect,

may be used. The local connections may include a mixture of various distance hops, such as distance 4 or 16 as well as distance 1. The outside edges may be connected using any topology, such as a torus or twisted torus. Instead of or in addition to a local interconnect, a more complex global interconnect, such as the hypercube design, may be used. Furthermore, the physical implementation of the PEA 104 (e.g., a chip) could be replicated (e.g., tiled on a circuit board) to produce a larger PEA. The replication may form a simple grid or other arrangement, just as the component PEAs may but need not be grids.

**[0054]** FIG. 4 shows an example design for a PE 400 (which may be used to implement any one or more of the PEs in the PEA 104). The PE 400 stores local data. The amount of memory for the local data varies significantly from design to design. It may depend on the implementation technologies available for fabricating the PE 400. Sometimes rarely changing values (Constants) take less room than frequently changing values (Registers), and a design may provide more Constants than Registers. For instance, this may be the case with digital embodiments that use single transistor cells for the Constants (e.g., floating gate Flash memory cells) and multiple transistor cells for the Registers (e.g., 6-transistor SRAM cells). Sometimes the situation is reversed, as may be the case in analog embodiments, where substantial area for capacitance may be needed to ensure stable long term storage of Constants, and such embodiments may have more Registers than Constants. Typical storage capacities might be tens or hundreds of arithmetic values stored in the Registers and Constants in each PE, but these capacities are adjustable by the designer. Some designs, for instance, may have Register storage but no Constant storage. Some designs may have thousands or even many more values stored in each PE. All of these variations may be reflected in embodiments of the present invention.

**[0055]** Each PE needs to operate on its local data. For this reason within the PE 400 there are data paths 402a-i, routing mechanisms (such as the multiplexor MUX 404), and components to perform some collection of logical and arithmetic operations (such as the logic unit 406 and the LPHDR arithmetic unit

408). The LPHDR arithmetic unit 408 performs LPHDR arithmetic operations, as that term is used herein. The input, output, and intermediate “values” received by, output by, and operated on by the PE 400 may, for example, take the form of electrical signals representing numerical values.

**[0056]** The PE 400 also may have one or more flag bits, shown as Mask 410 in FIG. 4. The purpose of the Mask 410 is to enable some PEs, the ones in which a specified Mask bit is set, to ignore some instructions issued by the CU 106. This allows some variation in the usual lock-step behaviors of all PEs in the PEA 104. For instance, the CU 106 may issue an instruction that causes each PE to reset or set its Mask 410 depending on whether a specified Register in the PE is positive or negative. A subsequent instruction, for instance an arithmetic instruction, may include a bit meaning that the instruction should be performed only by those PEs whose Mask 410 is reset. This combination has the effect of conditionally performing the arithmetic instruction in each PE depending on whether the specified Register in that PE was positive. As with the Compare instructions of traditional computers, there are many possible design choices for mechanisms to set and clear Masks.

**[0057]** The operation of the PEs is controlled by control signals 412a-d received from the CU 106, four of which are shown in FIG. 4 merely for purposes of example and not limitation. We have not shown details of this mechanism, but the control signals 412a-d specify which Register or Constant memory values in the PE 400 or one of its neighbors to send to the data paths, which operations should be performed by the Logic 406 or Arithmetic 408 or other processing mechanisms, where the results should be stored in the Registers, how to set, reset, and use the Mask 410, and so on. These matters are well described in the literature on SIMD processors.

**[0058]** Many variations of this PE 400 and PEA design are possible and fall within the scope of the present invention. Digital PEs can have shifters, lookup tables, and many other mechanisms such as described in the literature. Analog PEs can have time-based operators, filters, comparators with global broadcast signals and many other mechanisms such as described in the

Attorney Docket No. A0006-1001C2

literature. The PEA 104 can include global mechanisms such as wired-OR or wired-AND for digital PEAs or wired-SUM for analog PEAs. Again, there are many variations well described in the literature on digital and analog computing architectures.

**[0059]** For example, LPHDR operations other than and/or in addition to addition and multiplication may be supported. For example, a machine which can only perform multiplication and the function  $(1-X)$  may be used to approximate addition and other arithmetic operations. Other collections of LPHDR operations may be used to approximate LPHDR arithmetic operations, such as addition, multiplication, subtraction, and division, using techniques that are well-known to those having ordinary skill in the art.

**[0060]** One aspect of embodiments of the present invention that is unique is the inclusion of LPHDR arithmetic mechanisms in the PEs. Embodiments of such mechanisms will now be described.

**[0061]** One digital embodiment of the LPHDR arithmetic unit 408 operates on digital (binary) representations of numbers. In one digital embodiment these numbers are represented by their logarithms. Such a representation is called a Logarithmic Number System (LNS), which is well-understood by those having ordinary skill in the art.

**[0062]** In an LNS, numbers are represented as a sign and an exponent. There is an implicit base for the logarithms, typically 2 when working with digital hardware. In the present embodiment, a base of 2 is used for purposes of example. As a result, a value, say  $B$ , is represented by its sign and a base 2 logarithm, say  $b$ , of its absolute value. For numbers to have representation errors of at most, say, 1% (one percent), the fractional part of this logarithm should be represented with enough precision that the least possible change in the fraction corresponds to about a 1% change in the value  $B$ . If fractions are represented using 6 bits, increasing or decreasing the fraction by 1 corresponds to multiplying or dividing  $B$  by the 64th root of 2, which is approximately 1.011. This means that numbers may be represented in the present embodiment with a multiplicative error of approximately 1%. So, in this



Attorney Docket No. A0006-1001C2

example embodiment the fraction part of the representation has 6 bits.

**[0063]** Furthermore, the space of values processed in the present embodiment have high dynamic range. To represent numbers whose absolute value is from, say, one billionth to one billion, the integer part of the logarithm must be long enough to represent plus or minus the base 2 logarithm of one billion. That logarithm is about 29.9. In the present embodiment the integer part of the logarithm representation is 5 bits long to represent values from 0 to 31, which is sufficient. There also is a sign bit in the exponent. Negative logarithms are represented using two's complement representation.

**[0064]** In an LNS, the value zero corresponds to the logarithm negative infinity. One can choose a representation to explicitly represent this special value. However, to minimize resources (for instance, area) used by arithmetic circuits, the present embodiment represents zero by the most negative possible logarithm, which is -32, corresponding to the two's complement bit representation '100000 000000', and denoting a value of approximately  $2.33E-10$ .

**[0065]** When computing, situations can arise in which operations cannot produce reasonable values. An example is when a number is too large to be represented in the chosen word format, such as when multiplying or adding two large numbers or upon divide by zero (or nearly zero). One common approach to this problem is to allow a value to be marked as Not A Number (NAN) and to make sure that each operation produces NAN if a problem arises or if either of its inputs is NAN. The present embodiment uses this approach, as will be described in the following.

**[0066]** FIG. 5 shows the word format 500 for these numbers, in the present embodiment. It has one NAN bit 502a, one bit 502b for the sign of the value, and 12 bits 502c-e representing the logarithm. The logarithm bits include a 5 bit integer part 502d and a 6 bit fraction part 502e. To permit the logarithms to be negative, there is a sign bit 502c for the logarithm which is represented in two's complement form. The NAN bit is set if some problem has arisen in computing the value. The word format 500 shown in FIG. 5 is merely an example and does not constitute a limitation of the present invention. Other



Attorney Docket No. A0006-1001C2

variations may be used, so long as they have low precision and high dynamic range.

**[0067]** FIG. 6 shows an example digital implementation of the LPHDR arithmetic unit 408 for the representation illustrated in FIG. 5. The unit 408 receives two inputs, A 602a and B 602b, and produces an output 602c. The inputs 602a-b and output 602c may, for example, take the form of electrical signals representing numerical values according to the representation illustrated in FIG. 5, as is also true of signals transmitted within the unit 408 by components of the unit 408. The inputs 602a-b and output 602c each are composed of a Value and a NAN (Not A Number) bit. The unit 408 is controlled by control signals 412a-d, coming from the CU 106, that determine which available arithmetic operation will be performed on the inputs 602a-b. In this embodiment, all the available arithmetic operations are performed in parallel on the inputs 602a-b by adder/subtractor 604, multiplier 606, and divider 608. Adder/subtractor 604 performs LPHDR addition and subtraction, multiplier 606 performs LPHDR multiplication, and divider 608 performs LPHDR division.

**[0068]** The desired result (from among the outputs of adder/subtractor 604, multiplier 606, and divider 608) is chosen by the multiplexers (MUXes) 610a and 610b. The right hand MUX 610b sends the desired value to the output 602c. The left hand MUX 610a sends the corresponding NAN bit from the desired operation to the OR gate 612, which outputs a set NAN bit if either input is NAN or if the specified arithmetic operation yields NAN. The computing architecture literature discusses many variations which may be incorporated into the embodiment illustrated in FIG. 6.

**[0069]** LNS arithmetic has the great advantage that multiplication (MUL) and division (DIV) are very easy to compute and take few physical resources (e.g., little area in a silicon implementation). The sign of the result is the exclusive-or of the signs of the operands. The logarithm part of the output is the sum, in the case of MUL, or the difference, in the case of DIV, of the logarithm parts of the operands. The sum or difference of the logarithms can overflow, producing a NAN result. Certain other operations also are easy in LNS

Attorney Docket No. A0006-1001C2

arithmetic. For instance, square root corresponds to dividing the logarithm in half, which in our representation means simply shifting it one bit position to the right.

**[0070]** Thus, the multiplier 606 and divider 608 in FIG. 6 are implemented as circuits that simply add or subtract their inputs, which are two's complement binary numbers (which in turn happen to be logarithms). If there is overflow, they output a 1 for NAN.

**[0071]** Implementing addition and subtraction in LNS, that is, the adder/subtractor 604 in FIG. 6, follows a common approach used in the literature on LNS. Consider addition. If we have two positive numbers B and C represented by their logarithms b and c, the representation of the sum of B and C is  $\log(B+C)$ . An approach to computing this result that is well known to those skilled in the art is based on noticing that  $\log(B+C) = \log(B*(1+C/B)) = \log(B) + \log(1+C/B) = b + F(c-b)$  where  $F(x) = \log(1+2^x)$ . Thus, the present embodiment computes c-b, feeds that through F, and adds the result to b, using standard digital techniques known to those skilled in the art.

**[0072]** Much of the published literature about LNS is concerned with how to compute F(x), the special function for ADD, along with a similar function for SUB. Often these two functions share circuitry, and this is why a single combined adder/subtractor 604 is used in the embodiment of FIG. 6. There are many published ways to compute these functions or approximations to them, including discussions of how to do this when the values are of low precision. Any such method, or other method, may be used. Generally speaking, the more appropriate variations for massively parallel LPHDR arithmetic are those that require the minimal use of resources, such as circuit area, taking advantage of the fact that the representation used in the embodiment of FIG. 6 is low precision and that the arithmetic operations need not be deterministic nor return the most accurate possible answer within the low precision representation. Thus, embodiments of the present invention may use circuitry that does not compute the best possible answer, even among the limited choices available in a low precision representation.

**[0073]** In order to enable conditional operation of selected PEs, the present embodiment is able to reset and set the MASK flag 410 based on results of computations. The mechanism for doing this is that the CU 106 includes instructions that cause the MASK 410 in each PE to unconditionally reset or set its flag along with other instructions to perform basic tests on values entering the MASK 410 on data path 402f and to set the flag accordingly. Examples of these latter instructions include copying the sign bit or NAN bit of the word on data path 402f into the MASK bit 410. Another example is to set the MASK bit 410 if the 12 bit value part of the word on data path 402f is equal to binary zero. There are many additional and alternative ways for doing this that are directly analogous to comparison instructions in traditional processors and which are well understood by those skilled in the art.

**[0074]** It is worth noting that while the obvious method of using the above LNS operations is to do LPHDR arithmetic, the programmer also may consider selected values to be 12 bit two's complement binary numbers. MUL and DIV may be used to add and subtract such values, since that is precisely their behavior in LNS implementations. The Mask setting instructions can compare these simple binary values. So besides doing LPHDR computations, this digital embodiment using LNS can perform simple binary arithmetic on short signed integers.

**[0075]** Some embodiments of the present invention may include analog representations and processing methods. Such embodiments may, for example, represent LPHDR values as charges, currents, voltages, frequencies, pulse widths, pulse densities, various forms of spikes, or in other forms not characteristic of traditional digital implementations. There are many such representations discussed in the literature, along with mechanisms for processing values so represented. Such methods, often called Analog methods, can be used to perform LPHDR arithmetic in the broad range of architectures we have discussed, of which SIMD is one example.

**[0076]** An example of an analog SIMD architecture is the SCAMP design (and related designs) of Dudek. In that design values have low dynamic

range, being accurate roughly to within 1%. Values are represented by charges on capacitors. Those capacitors typically are the gates of transistors. Each PE has several memory cells, analogous to the Registers shown in FIG. 4. Addition is performed by turning on pass transistors from the two operands, which transfer their charge onto an analog bus, where it is summed by the natural physics of charge and wires, upon which it is gated onto another Register to charge up its capacitor, which then represents the sum of the operands. The detailed mechanism disclosed by Dudek actually produces the negative of the sum, but the basic concept is as described and is a simple way to perform addition and subtraction using analog representations and simple processing mechanisms.

**[0077]** Variations of the SCAMP design have been fabricated and used to perform a range of low precision, low dynamic range computations related to image processing. These designs do not perform high dynamic range arithmetic, nor do they include mechanisms for performing multiplication or division of values stored in Registers. However, the Dudek designs suggest the general feasibility of constructing analog SIMD machines. The following describes how to build an analog SIMD machine that performs LPHDR arithmetic, and is thus an embodiment of the present invention.

**[0078]** One embodiment of the present invention represents values as a mixture of analog and digital forms. This embodiment represents values as low precision, normalized, base 2 floating point numbers, where the mantissa is an analog value and the exponent is a binary digital value. The analog value may be accurate to about 1%, following the approach of Dudek, which is well within the range of reasonable analog processing techniques. The exponent may be 6 bits long, or whatever is needed to provide the desired high dynamic range.

**[0079]** To multiply values, the embodiment proceeds by analogy to traditional floating point methods. The digital exponents are summed using a binary arithmetic adder, a standard digital technique. The analog mantissas are multiplied. Since they represent normalized values between approximately  $1/2$  and 1, their product may be as small as approximately  $1/4$ . Such a product value needs to be normalized back to the range  $1/2$  to 1. This is done, in the present

Attorney Docket No. A0006-1001C2

embodiment, by comparing the analog mantissa to an analog representation of  $1/2$ , using a threshold circuit. If the mantissa is below  $1/2$ , then it is doubled and one is subtracted from the exponent, where such subtraction is simple digital subtraction. Doubling the mantissa is implemented in a way that corresponds to the chosen analog representation. For example, whatever means are being used to add two analog values can be used to double the mantissa, by adding it to a copy of itself. For example, if the mantissa is represented as a current, such as copy may be produced by a current mirror, or other suitable mechanism, and addition may be performed by a current summing junction.

**[0080]** The means of multiplying the original analog mantissas depends on the representation chosen. For example, if mantissas are represented using charge, following SCAMP, then any known method from the literature may be used to convert charge to current. For instance, since the charge on a capacitor determines the voltage on the capacitor, this may be implemented as a conversion from voltage to current, which is a basic technique in analog electronics known to those skilled in the art. In any case, if the mantissas are represented as currents, or once the mantissas are converted to currents, they may be multiplied using, for instance, the techniques of Gilbert. The Gilbert multiplier produces a current, representing the product, which may, if necessary, then be converted back to charge (or whatever representation is used). These are merely examples of how the needed operations might be performed. The literature discusses these matters extensively and these kinds of analog circuits are known to those skilled in the art.

**[0081]** Adding and subtracting values requires pre-normalization of the values to the same exponent, as is done in traditional digital floating point arithmetic. The present embodiment does this by comparing the exponents and choosing the smaller one. Then the smaller one is subtracted from the larger, using digital means. The difference specifies how many times the mantissa which corresponds to the smaller exponent needs to be divided in half. If that mantissa is represented by (or converted to) a current, then an analog R-2R style ladder may be used to divide the current in half the required number of times,



Attorney Docket No. A0006-1001C2

with the stage of the ladder specified by the difference of exponents calculated as above. The resulting scaled down current is added to (or subtracted from, if this is an LPHDR subtraction operation) the current corresponding to the mantissa associated with the larger exponent to yield the output mantissa. The output exponent associated with the output mantissa is the larger exponent. Post-normalization may be needed at this point. If the output mantissa is greater than 1, then it needs to be divided in half and the output exponent needs to be incremented. If it is less than  $1/2$ , then it needs to be doubled enough times to exceed  $1/2$  and the output exponent must be decremented correspondingly, which may be performed by a series of threshold circuits, doubler circuits, and associated decrementer circuits. These increments and decrements of the binary digital exponent, and corresponding doublings and halvings of the analog mantissa current, are straightforward operations well known to those skilled in the art.

**[0082]** The present embodiment represents the exponent as a digital binary number. Alternate embodiments may represent the exponent as an analog value. However, it is important that the exponent be represented in storage and computation in such a manner that neither noise nor other errors cause a change in the value it represents. Such changes in the exponent could introduce factors of two (or in some embodiments larger) changes in the values of the stored numbers. To maintain accuracy of the exponents, an embodiment may quantize the exponent to relatively few levels, for instance 16 values plus a sign bit. During processing, slight variations in the analog representation of the exponent may then be removed by circuitry that restores values to the 16 standard quantization levels. To get sufficient dynamic range in such an embodiment, the floating point numbers may be processed as base 4 numbers, rather than the usual base 2 numbers. This means, for instance, that normalized mantissas are in the range  $1/4$  to 1. The methods discussed above for addition, subtraction, and multiplication apply as described, with slight and straightforward variations.

**[0083]** The analog and mixed signal embodiments discussed above

are merely examples and do not constitute a limitation of the present invention. The published literature on neuromorphic, analog, and mixed signal techniques provides a wealth of methods that enable LPHDR storage and processing to be implemented. Such storage and processing may introduce noise as well as fabrication errors into the behavior of machines performing LPHDR arithmetic. The results we present below, on software applications running using “fp+noise” arithmetic, show that despite these very “un-digital” qualities a machine built in this way is surprisingly useful.

**[0084]** Evidence that LPHDR arithmetic is useful in several important practical computing applications will now be provided. The evidence is presented for a broad variety of embodiments of the present invention, thereby showing that the usefulness does not depend much on the detailed implementation.

**[0085]** For the goal of showing usefulness, we choose a very general embodiment of an LPHDR machine. Our model of the machine is that it provides at least the following capabilities: (1) is massively parallel, (2) provides LPHDR arithmetic possibly with noise, (3) provides a small amount of memory local to each arithmetic unit, (4) provides the arithmetic/memory units in a two-dimensional physical layout with only local connections between units (rather than some more powerful, flexible, or sophisticated connection mechanism), and (5) provides only limited bandwidth between the machine and the host machine. Note that this model is merely an example which is used for the purpose of demonstrating the utility of various embodiments of the present invention, and does not constitute a limitation of the present invention. This model includes, among others, implementations that are digital or analog or mixed, have zero or more noise, have architectures which are FPGA-like, or SIMD-like, or MIMD-like, or otherwise meet the assumptions of the model. More general architectures, such as shared memory designs, GPU-like designs, or other sophisticated designs subsume this model's capabilities, and so LPHDR arithmetic in those architectures also is useful. While we are thus showing that LPHDR arithmetic is useful for a broad range of designs, of which SIMD is only an instance, for

Attorney Docket No. A0006-1001C2

purpose of discussion below, we call each unit, which pairs memory with arithmetic, a Processing Element or "PE".

**[0086]** Several applications are discussed below. For each, the discussion shows (1) that the results are useful when computation is performed in possibly noisy LPHDR arithmetic, and (2) that the computation can be physically laid out in two dimensions with only local flow of data between units, only limited memory within each unit, and only limited data flow to/from the host machine, in such a way that the computation makes efficient use of the machine's resources (area, time, power). The first requirement is referred to as "Accuracy" and the second requirement "Efficiency." Applications that meet both requirements running in this model will function well on many kinds of LPHDR machines, and thus those machines are a broadly useful invention.

**[0087]** Applications are tested using two embodiments for the machine's arithmetic. One uses accurate floating point arithmetic but multiplies the result of each arithmetic operation by a uniformly chosen random number between .99 and 1.01. In the following discussion, this embodiment is denoted "fp+noise". It may represent the results produced by an analog embodiment of the machine.

**[0088]** A second embodiment uses logarithmic arithmetic with a value representation as shown in FIG. 5. The arithmetic is repeatable, that is, not noisy, but because of the short fraction size it produces errors of up to approximately 1-2% in each operation. In the following discussion, this embodiment is denoted "lns". It may represent the results produced by a particular digital embodiment of the machine.

**[0089]** To demonstrate usefulness of embodiments of the invention, we shall discuss three computational tasks that are enabled by embodiments of the invention and which in turn enable a variety of practical applications. Two of the tasks are related to finding nearest neighbors and the other is related to processing visual information. We shall describe the tasks, note their practical application, and then demonstrate that each task is solvable using the general model described above and thus solvable using embodiments of the present

invention.

Application 1: Finding Nearest Neighbors

**[0090]** Given a large set of vectors, called Examples, and a given vector, called Test, the nearest neighbor problem ("NN") is to find the Example which is closest to Test where the distance metric is the square of the Euclidean distance (sum of squares of distances between respective components).

**[0091]** NN is a widely useful computation. One use is for data compression, where it is called "vector quantization". In this application we have a set of relatively long vectors in a "code book" (these are the Examples) and associated short code words (for instance the index of the vector in the code book). We move through a sequence of vectors to be compressed, and for each such vector (Test), find the nearest vector in the code book and output the corresponding code word. This reduces the sequence of vectors to the shorter sequence of code words. Because the code words do not completely specify the original sequence of vectors, this is a lossy form of data compression. Among other applications, it may be used in speech compression and in the MPEG standards.

**[0092]** Another application of NN would be in determining whether snippets of video occur in a large video database. Here we might abstract frames of video from the snippet into feature vectors, using known methods, such as color histograms, scale invariant feature extraction, etc. The Examples would be analogous feature vectors extracted from the video database. We would like to know whether any vector from the snippet was close to any vector from the database, which NN can help us decide.

**[0093]** In many applications of nearest neighbor, we would prefer to find the true nearest neighbor but it is acceptable if we sometimes find another neighbor that is only slightly farther away or if we almost always find the true nearest neighbor. Thus, an approximate solution to the nearest neighbor problem is useful, especially if it can be computed especially quickly, or at low power, or with some other advantage compared to an exact solution.

**[0094]** We shall now show that approximate nearest neighbor is

Attorney Docket No. A0006-1001C2

computable using embodiments of the present invention in a way that meets the criteria of Accuracy and Efficiency.

**[0095]**     Algorithm. The following describes an algorithm which may be performed by machines implemented according to embodiments of the present invention, such as by executing software including instructions for performing the algorithm. The inputs to the algorithm are a set of Examples and a Test vector. The algorithm seeks to find the nearest (or almost nearest) Example to the Test.

**[0096]**     In the simplest version of the algorithm, the number of Examples may be no larger than the number of PEs and each vector must be short enough to fit within a single PE's memory. The Examples are placed into the memories associated with the PEs, so that one Example is placed in each PE. Given a Test, the Test is passed through all the PEs, in turn. Accompanying the Test as it passes through the PEs is the distance from the Test to the nearest Example found so far, along with information that indicates what PE (and thus what Example) yielded that nearest Example found so far. Each PE computes the distance between the Test and the Example stored in that PE's memory, and then passes along the Test together with either the distance and indicator that was passed into this PE (if the distance computed by this PE exceeded the distance passed into the PE) or the distance this PE computed along with information indicating this PE's Example is the nearest so far (if the distance computed by this PE is less than the distance passed into the PE). Thus, the algorithm is doing a simple minimization operation as the Test is passed through the set of PEs. When the Test and associated information leave the last PE, the output is a representation of which PE (and Example) was closest to the Test, along with the distance between that Example and the Test.

**[0097]**     In a more efficient variant of this algorithm, the Test is first passed along, for example, the top row, then every column passes the Test and associated information downward, effectively doing a search in parallel with other columns, and once the information reaches the bottom it passes across the bottom row computing a minimum distance Example of all the columns processed so far as it passes across the row. This means that the time required



Attorney Docket No. A0006-1001C2

to process the Test is proportional to (the greater of) the number of PEs in a row or column.

**[0098]** An enhancement of this algorithm proceeds as above but computes and passes along information indicating both the nearest and the second nearest Example found so far. When this information exits the array of PEs, the digital processor that is hosting the PE array computes (in high precision) the distance between the Test and the two Examples indicated by the PE array, and the nearer of the two is output as the likely nearest neighbor to the Test.

**[0099]** Accuracy. We expressed the arithmetic performed by the enhanced algorithm described above as code in the C programming language. That code computes both nearest neighbors, which are discussed here, along with weighted scores, which are discussed below.

**[0100]** The C code performs the same set of arithmetic operations in the same order using the same methods of performing arithmetic as an actual implementation of the present invention, such as one implemented in hardware. It thus yields the same results as the enhanced algorithm would yield when running on an implementation of the present invention. (How the algorithm is organized to run efficiently on such an implementation is discussed below in the section on Efficiency.)

**[0101]** In particular, when computing the distance between the Test and each Example, the code uses Kahan's method, discussed below, to perform the possibly long summation required to form the sum of the squares of the distances between vector components of the Test and Example.

**[0102]** The C code contains several implementations for arithmetic, as discussed above. When compiled with "#define fp" the arithmetic is done using IEEE standard floating point. If a command line argument is passed in to enable noisy arithmetic, then random noise is added to the result of every calculation. This is the "fp+noise" form of arithmetic. When compiled without "#define fp" the arithmetic is done using low precision logarithmic arithmetic with a 6 bit base-2 fraction. This is the "lns" form of arithmetic.

Attorney Docket No. A0006-1001C2

**[0103]** When the code was run it produced traces showing the results of the computations it performed. These traces, shown below, show that with certain command line arguments the enhanced algorithm yielded certain results for LPHDR nearest neighbor calculations. These results provide details showing the usefulness of this approach. We shall discuss the results briefly here.

**[0104]** The first results are for "fp+noise". Ten distinct runs were performed. Each run generated one million random Example vectors of length five, where each component of each vector was drawn from  $N(0,1)$ —the Gaussian (normal) distribution with mean zero and standard deviation 1. Each run then generated one hundred Test vectors of length five, where each component of each vector also was drawn from  $N(0,1)$ . For each Test, the nearest neighbor was computed both according to the enhanced algorithm above and according to the standard nearest neighbor method using high precision floating point arithmetic. A count was kept of the number of times the enhanced algorithm yielded the same result as the standard floating point method. The results were as follows:

```
% ./a.out 5 10 1000000 100 1
Representation is Floating Point with noise.
Run 1. On 100 tests, 100(100.0%) matches and 0.81% mean score error.
Run 2. On 100 tests, 100(100.0%) matches and 0.84% mean score error.
Run 3. On 100 tests, 100(100.0%) matches and 0.98% mean score error.
Run 4. On 100 tests, 100(100.0%) matches and 0.81% mean score error.
Run 5. On 100 tests, 100(100.0%) matches and 0.94% mean score error.
Run 6. On 100 tests, 100(100.0%) matches and 0.82% mean score error.
Run 7. On 100 tests, 100(100.0%) matches and 0.78% mean score error.
Run 8. On 100 tests, 100(100.0%) matches and 0.86% mean score error.
Run 9. On 100 tests, 100(100.0%) matches and 0.85% mean score error.
Run 10. On 100 tests, 99(99.0%) matches and 0.86% mean score error.
Average percentage of time LPHDR (with final DP correction) finds
nearest example = 99.90%.
Average score error between LPHDR and DP = 0.85%.
```

**[0105]** The "mean score error" values are considered below in the discussion of weighted scores. The "matches" information is relevant here.

**[0106]** Of the ten runs, only one had any test, of the 100 tests performed, which yielded a nearest neighbor different from what the usual high

Attorney Docket No. A0006-1001C2

precision method yielded. Thus, the average percentage of matches between the enhanced algorithm running with "fp+noise" arithmetic and the usual method was 99.9%.

**[0107]** A similar computation was then performed using "lns" arithmetic. In this case, the results were:

```
% ./a.out 5 10 1000000 100 0
```

Representation is LNS without noise.

Run 1. On 100 tests, 100(100.0%) matches and 0.15% mean score error.

Run 2. On 100 tests, 100(100.0%) matches and 0.07% mean score error.

Run 3. On 100 tests, 100(100.0%) matches and 0.08% mean score error.

Run 4. On 100 tests, 100(100.0%) matches and 0.09% mean score error.

Run 5. On 100 tests, 100(100.0%) matches and 0.11% mean score error.

Run 6. On 100 tests, 100(100.0%) matches and 0.16% mean score error.

Run 7. On 100 tests, 100(100.0%) matches and 0.07% mean score error.

Run 8. On 100 tests, 100(100.0%) matches and 0.13% mean score error.

Run 9. On 100 tests, 99(99.0%) matches and 0.17% mean score error.

Run 10. On 100 tests, 98(98.0%) matches and 0.16% mean score error.

Average percentage of time LPHDR (with final DP correction) finds nearest example = 99.70%.

Average score error between LPHDR and DP = 0.12%.

**[0108]** The average percentage of matches was 99.7%, slightly worse than for "fp+noise".

**[0109]** The accuracy shown by the enhanced nearest neighbor algorithm using two forms of LPHDR arithmetic is surprising. To perform many calculations sequentially with 1% error and yet produce a final result with less than 1% error may seem counter-intuitive. Nonetheless, the LPHDR arithmetic proves effective, and the accuracy shown is high enough to be useful in applications for which approximate nearest neighbor calculations are useful.

**[0110]** As an extreme case, a variant of fp+noise was tested in which the noise varied uniformly from +10% to -5%. Thus, each arithmetic operation produced a result that was between 10% too large and 5% too small. The enhanced nearest neighbor algorithm, as described above, was performed where each run generated 100,000 Example vectors. The surprising results, below, show that even with this extreme level of imprecise, noisy, and non-zero mean LPHDR arithmetic, useful results can be achieved.

Attorney Docket No. A0006-1001C2

Run 1. On 100 tests, 97(97.0%) matches.  
 Run 2. On 100 tests, 100(100.0%) matches.  
 Run 3. On 100 tests, 100(100.0%) matches.  
 Run 4. On 100 tests, 98(98.0%) matches.  
 Run 5. On 100 tests, 98(98.0%) matches.  
 Run 6. On 100 tests, 99(99.0%) matches.  
 Run 7. On 100 tests, 99(99.0%) matches.  
 Run 8. On 100 tests, 99(99.0%) matches.  
 Run 9. On 100 tests, 99(99.0%) matches.  
 Run 10. On 100 tests, 99(99.0%) matches.  
 Average percentage of time LPHDR (with final DP correction) finds nearest example = 98.80%.

**[0111]** Efficiency. In contrast to the surprising Accuracy results, it is clear to those having ordinary skill in the art that the calculations of the enhanced nearest neighbor algorithm can be performed efficiently in the computing model presented, where the arithmetic/memory units are connected in a two-dimensional physical layout, using only local communication between PEs. However, this does not address the matter of keeping the machine busy doing useful work using only low bandwidth to the host machine.

**[0112]** When computing the nearest neighbor to a single Test, the Test flows across all the PEs in the array. As discussed above, if the array is an  $M \times M$  grid, it takes at least  $O(M)$  steps for the Test to pass through the machine and return results to the host. During this time the machine performs  $O(M \times M)$  nearest neighbor distance computations, but since the machine is capable of performing  $O(M \times M)$  calculations at each step, a factor of  $O(M)$  is lost.

**[0113]** This speedup, compared to a serial machine, of a factor of  $O(M)$  is significant and useful. However, the efficiency can be even higher. If sufficiently many Test vectors, say  $O(M)$ , or more, are to be processed then they can be streamed into the machine and made to flow through in a pipelined fashion. The time to process  $O(M)$  Tests remains  $O(M)$ , the same as for a single Test, but now the machine performs  $O(M) \times O(M \times M)$  distance computations, and thus within a constant factor the full computing capacity of the machine is used.

**[0114]** Thus, the machine is especially efficient if it is processing at least as many Test vectors as the square root of the number of PEs. There are

Attorney Docket No. A0006-1001C2

applications that fit well into this form, such as pattern recognition or compression of many independent Tests (e.g., blocks of an image, parts of a file, price histories of independent stocks) as well as the problem of finding the nearest neighbor to every Example in the set of Examples. This is in contrast to the general view among those having ordinary skill in the art, as discussed above, that machines with very many arithmetic processing elements on a single chip, or similar, are not very useful.

#### Application 2: Distance Weighted Scoring

**[0115]** A task related to Nearest Neighbor is Distance Weighted Scoring. In this task, each Example has an associated Score. This is a number that in some way characterizes the Example. For instance, if the Examples are abstractions of the history of prices of a given stock, the Scores might be historical probabilities of whether the price is about to increase or decrease. Given a Test vector, the task is to form a weighted sum of the Scores of all the Examples, where the weights are a diminishing function of the distance from the Test to the respective Examples. For example, this weighted score might be taken as a prediction of the future price of the stock whose history is represented by the Test. This use of embodiments of the invention might help support, for instance, high speed trading of stocks, as is performed by certain "quantitative" hedge funds, despite the general view by those having ordinary skill in the art that low precision computation is not of use in financial applications.

**[0116]** The C code described above computes weighted scores along with nearest neighbors. The scores assigned to Examples in this computation are random numbers drawn uniformly from the range [0,1]. The weight for each Example in this computation is defined to be the un-normalized weight for the Example divided by the sum of the un-normalized weights for all Examples, where the un-normalized weight for each Example is defined to be the reciprocal of the sum of one plus the squared distance from the Example to the Test vector. As discussed above, the code performs a number of runs, each producing many Examples and Tests, and compares results of traditional floating point computations with results calculated using fp+noise and Ins arithmetic.



Attorney Docket No. A0006-1001C2

**[0117]** Looking again at the trace results of running the simulation, above, we see that for fp+noise the LPHDR weighted scores on average were within .85% of the correct value and never were as much as 1% different. For Ins arithmetic the errors were even smaller, averaging just .12% error.

**[0118]** These results are surprising given that computing an overall weighted score involves summing the individual weighted scores associated with each Example. Since each run was processing 1,000,000 Examples, this means that the sums were over one million small positive values. The naive method of summing one million small values with errors of about 1% in each addition should yield results that approximate noise. However, the code performs its sums using a long known method invented by Kahan (Kahan, William (January 1965), "Further remarks on reducing truncation errors", Communications of the ACM 8 (1): 40). The method makes it feasible to perform long sums, such as are done for Distance Weighted Scores, or as might be used in computational finance when computing prices of derivative securities using Monte Carlo methods, or for performing deconvolution in image processing algorithms, as will be discussed next.

**[0119]** The Efficiency of this algorithm is similar to that of NN, as discussed earlier. If many Test vectors are processed at once, the machine performs especially efficiently.

#### Application 3: Removing motion blur in images

**[0120]** In order to gather sufficient light to form an image, camera shutters are often left open for long enough that camera motion can cause blurring. This can happen as a result of camera shake in inexpensive consumer cameras as well as with very expensive but fast moving cameras mounted on satellites or aircraft. If the motion path of the camera is known (or can be computed) then the blur can be substantially removed using various deblurring algorithms. One such algorithm is the Richardson-Lucy method ("RL"), and we show here that embodiments of the present invention can run that algorithm and produce useful results. Following the discussion format above, we discuss criteria of Accuracy and Efficiency.

**[0121]** Algorithm. The Richardson-Lucy algorithm is well known and widely available. Assume that an image has been blurred using a known kernel. In particular, assume that the kernel is a straight line and that the image has been oriented so that the blur has occurred purely in a horizontal direction. Consider the particular kernel for which the J'th pixel in each row of the blurred image is the uniformly weighted mean of pixels J through J+31 in the original unblurred image.

**[0122]** Accuracy. We implemented in the C programming language a straightforward version of the RL method that uses LPHDR arithmetic. The program reads a test image, blurs it using the kernel discussed above, then deblurs it using either fp+noise or lns arithmetic. The RL algorithm computes sums, such as when convolving the kernel with the current approximation of the deblurred image. Our implementation computes these sums using the Kahan method, discussed earlier. FIG. 7 shows the test image in original form. It is a satellite picture of a building used during Barack Obama's inauguration. FIG. 8 shows the image extremely blurred by the kernel. It is difficult to see any particular objects in this image. FIG. 9 shows the result of deblurring using standard floating point arithmetic. FIG. 10 shows the result of deblurring using fp+noise arithmetic, and FIG.11 shows the result of deblurring using lns arithmetic. In all these cases the image is sufficiently restored that it is possible to recognize buildings, streets, parking lots, and cars.

**[0123]** In addition to displaying the images herein for judgement using the human eye, we computed a numerical measure of deblurring performance. We computed the mean difference, over all pixels in the image, between each original pixel value (a gray scale value from 0 to 255) and the corresponding value in the image reconstructed by the RL method. Those numerical measures are shown below in Table 1:

<u>Image type</u>	<u>Mean pixel error</u>
Blurred	32.0
RL using standard floating point	13.0
RL using fp+noise	13.8
RL using lns	14.8

Table 1

**[0124]** These results, together with the subjective but important judgements made by the human eye, show that LPHDR arithmetic provides a substantial and useful degree of deblurring compared to standard floating point arithmetic. Further, in this example we chose an extreme degree of blurring, to better convey the concept and visual impact of the deblurring using LPHDR arithmetic. On more gentle and typical blur kernels, the resulting deblurred images are much closer to the originals than in this case, as can be seen by shrinking the kernel length and running the RL algorithm with LPHDR arithmetic on those more typical cases.

**[0125]** Efficiency. It is clear to those with ordinary skill in the art that Richardson-Lucy using a local kernel performs only local computational operations. An image to be deblurred can be loaded into the PE array, storing one or more pixels per PE, the deconvolution operation of RL can then be iterated dozens or hundreds of times, and the deblurred image can be read back to the host processor. As long as sufficient iterations are performed, this makes efficient use of the machine.

**[0126]** An extreme form of image deblurring is the Iterative Reconstruction method used in computational tomography. Reconstructing 3D volumes from 2D projections is an extremely computational task. The method discussed above generalizes naturally to Iterative Reconstruction and makes efficient use of the machine.

**[0127]** Among the advantages of embodiments of the invention are one or more of the following.

**[0128]** PEs implemented according to certain embodiments of the present invention may be relatively small for PEs that can do arithmetic. This means that there are many PEs per unit of resource (e.g., transistor, area, volume), which in turn means that there is a large amount of arithmetic computational power per unit of resource. This enables larger problems to be solved with a given amount of resource than does traditional computer designs. For instance, a digital embodiment of the present invention built as a large silicon

chip fabricated with current state of the art technology might perform tens of thousand of arithmetic operations per cycle, as opposed to hundreds in a conventional GPU or a handful in a conventional multicore CPU. These ratios reflect an architectural advantage of embodiments of the present invention that should persist as fabrication technology continues to improve, even as we reach nanotechnology or other implementations for digital and analog computing.

**[0129]** Doing arithmetic with few resources generally means, and in the embodiments shown specifically means, that the arithmetic is done using low power. As a result, a machine implemented in accordance with embodiments of the present invention can have extremely high performance with reasonable power (for instance in the tens of watts) or low power (for instance a fraction of a watt) with reasonably high performance. This means that such embodiments may be suitable for the full range of computing, from supercomputers, through desktops, down to mobile computing. Similarly, and since cost is generally associated with the amount of available resources, embodiments of the present invention may provide a relatively high amount of computing power per unit of cost compared to conventional computing devices.

**[0130]** The SIMD architecture is rather old and is frequently discarded as an approach to computer design by those having ordinary skill in the art. However, if the processing elements of a SIMD machine can be made particularly small while retaining important functionality, such as general arithmetic ability, the architecture can be useful. The embodiments presented herein have precisely those qualities.

**[0131]** The discovery that massive amounts of LPHDR arithmetic is useful as a fairly general computing framework, as opposed to the common belief that it is not useful, can be an advantage in any (massively or non-massively) parallel machine design or non-parallel design, not just in SIMD embodiments. It could be used in FPGAs, FPAAs, GPU/SIMT machines, MIMD machines, and in any kind of machine that uses compact arithmetic processing elements to perform large amounts of computation using a small amount of resources (like transistors or volume).

**[0132]** Another advantage of embodiments of the present invention is that they are not merely useful for performing computations efficiently in general, but that they can be used to tackle a variety of real-world problems which are typically assumed to require high-precision computing elements, even though such embodiments include only (or predominantly) low-precision computing elements. Although several examples of such real-world problems have been presented herein, and although we have also had success implementing non-bonded force field computations for molecular dynamics simulation and other tasks, these are merely examples and do not constitute an exhaustive set of the real-world problems that embodiments of the present invention may be used to solve.

**[0133]** The embodiments disclosed above are merely examples and do not constitute limitations of the present invention. Rather, embodiments of the present invention may be implemented in a variety of other ways, such as the following.

**[0134]** For example, embodiments of the present invention may represent values in any of a variety of ways, such as by using digital or analog representations, such as fixed point, logarithmic, or floating point representations, voltages, currents, charges, pulse width, pulse density, frequency, probability, spikes, timing, or combinations thereof. These underlying representations may be used individually or in combination to represent the LPHDR values. LPHDR arithmetic circuits may be implemented in any of a variety of ways, such as by using various digital methods (which may be parallel or serial, pipelined or not) or analog methods or combinations thereof. Arithmetic elements may be connected using various connection architectures, such as nearest 4, nearest 8, hops of varying degree, and architectures which may or may not be rectangular or grid-like. Any method may be used for communication among arithmetic elements, such as parallel or serial, digital or analog or mixed-mode communication. Arithmetic elements may operate synchronously or asynchronously, and may operate globally simultaneously or not. Arithmetic elements may be implemented, for example, on a single physical device, such as a silicon chip, or



spread across multiple devices and an embodiment built from multiple devices may have its arithmetic elements connected in a variety of ways, including for example being connected as a grid, torus, hypercube, tree, or other method. Arithmetic elements may be connected to a host machine, if any, in a variety of ways, depending on the cost and bandwidth and other requirements of a particular embodiment. For example there may be many host machines connected to the collection of arithmetic elements.

**[0135]** Although certain embodiments of the present invention are described as being implemented as a SIMD architecture, this is merely an example and does not constitute a limitation of the present invention. For example, embodiments of the present invention may be implemented as reconfigurable architectures, such as but not limited to programmable logic devices, field programmable analog arrays, or field programmable gate array architectures, such as a design in which existing multiplier blocks of an FPGA are replaced with or supplemented by LPHDR arithmetic elements of any of the kinds disclosed herein, or for example in which LPHDR elements are included in a new or existing reconfigurable device design. As another example, embodiments of the present invention may be implemented as a GPU or SIMT-style architecture which incorporates LPHDR arithmetic elements of any of the kinds disclosed herein. For example, LPHDR elements could supplement or replace traditional arithmetic elements in current or new graphics processing unit designs. As yet another example, embodiments of the present invention may be implemented as a MIMD-style architecture which incorporates LPHDR arithmetic elements of any of the kinds disclosed herein. For example, LPHDR arithmetic elements could supplement or replace traditional arithmetic elements in current or new MIMD computing system designs. As yet another example, embodiments of the present invention may be implemented as any kind of machine, including a massively parallel machine, which uses compact arithmetic processing elements to provide large amounts of arithmetic computing capability using a small amount of resources (for example, transistors or area or volume) compared with traditional architectures.

**[0136]** Although certain embodiments of the present invention are described herein as executing software, this is merely an example and does not constitute a limitation of the present invention. Alternatively, for example, embodiments of the present invention may be implemented using microcode or a hardware sequencer or state machine or other controller to control LPHDR arithmetic elements of any of the kinds disclosed herein. Alternatively, for example, embodiments of the present invention may be implemented using hardwired, burned, or otherwise pre-programmed controllers to control LPHDR arithmetic elements of any of the kinds disclosed herein.

**[0137]** Although certain embodiments of the present invention are described herein as being implemented using custom silicon as the hardware, this is merely an example and does not constitute a limitation of the present invention. Alternatively, for example, embodiments of the present invention may be implemented using FPGA or other reconfigurable chips as the underlying hardware, in which the FPGAs or other reconfigurable chips are configured to perform the LPHDR operations disclosed herein. As another example, embodiments of the present invention may be implemented using any programmable conventional digital or analog computing architecture (including those which use high-precision computing elements, including those which use other kinds of non-LPHDR hardware to perform LPHDR arithmetic, and including those which are massively parallel) which has been programmed with software to perform the LPHDR operations disclosed herein. For example, embodiments of the present invention may be implemented using a software emulator of the functions disclosed herein.

**[0138]** As yet another example, embodiments of the present invention may be implemented using 3D fabrication technologies, whether based on silicon chips or otherwise. Some example embodiments are those in which a memory chip has been bonded onto a processor or other device chip or in which several memory and/or processor or other device chips have been bonded to each other in a stack. 3D embodiments of the present invention are very useful as they may be denser than 2D embodiments and may enable 3D communication of

information between the processing units, which enables more algorithms to run efficiently on those embodiments compared to 2D embodiments.

**[0139]** Although certain embodiments of the present invention are described herein as being implemented using silicon chip fabrication technology, this is merely an example and does not constitute a limitation of the present invention. Alternatively, for example, embodiments of the present invention may be implemented using technologies that may enable other sorts of traditional digital and analog computing processors or other devices. Examples of such technologies include various nanomechanical and nanoelectronic technologies, chemistry based technologies such as for DNA computing, nanowire and nanotube based technologies, optical technologies, mechanical technologies, biological technologies, and other technologies whether based on transistors or not that are capable of implementing LPHDR architectures of the kinds disclosed herein.

**[0140]** Certain embodiments of the present invention have been described as “massively parallel” embodiments. Although certain embodiments of the present invention may include thousands, millions, or more arithmetic units, embodiments of the present invention may include any number of arithmetic units (as few as one). For example, even an embodiment which includes only a single LPHDR unit may be used within a serial processing unit or other device to provide a significant amount of LPHDR processing power in a small, inexpensive processor or other device.

**[0141]** For certain embodiments of the present invention, even if implemented using only digital techniques, the arithmetic operations may not yield deterministic, repeatable, or the most accurate possible results within the chosen low precision representation. For instance, on certain specific input values, an arithmetic operation may produce a result which is not the nearest value in the chosen representation to the true arithmetic result.

**[0142]** The degree of precision of a “low precision, high dynamic range” arithmetic element may vary from implementation to implementation. For example, in certain embodiments, a LPHDR arithmetic element produces results

which include fractions, that is, values greater than zero and less than one. For example, in certain embodiments, a LPHDR arithmetic element produces results which are sometimes (or all of the time) no closer than 0.05% to the correct result (that is, the absolute value of the difference between the produced result and the correct result is no more than one-twentieth of one percent of the absolute value of the correct result). As another example, a LPHDR arithmetic element may produce results which are sometimes (or all of the time) no closer than 0.1% to the correct result. As another example, a LPHDR arithmetic element may produce results which are sometimes (or al of the time) no closer than 0.2% to the correct result. As yet another example, a LPHDR arithmetic element may produce results which are sometimes (or all of the time) no closer than 0.5% to the correct result. As yet further examples, a LPHDR arithmetic element may produce results which are sometimes (or all of the time) no closer than 1%, or 2%, or 5%, or 10%, or 20% to the correct result.

**[0143]** Besides having various possible degrees of precision, implementations may vary in the dynamic range of the space of values they process. For example, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range approximately from one millionth to one million. As another example, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range approximately from one billionth to one billion. As yet another example, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range approximately from one sixty five thousandth to sixty five thousand. As yet further examples, in certain embodiments, a LPHDR arithmetic element processes values in a space which may range from any specific value between zero and one sixty five thousandth up to any specific value greater than sixty five thousand. As yet further examples, other embodiments may process values in spaces with dynamic ranges that may combine and may fall between the prior examples, for example ranging from approximately one billionth to ten million. In all of these example embodiments of the present invention, as well as in other embodiments, the values that we are discussing may be signed, so that the

Attorney Docket No. A0006-1001C2

above descriptions characterize the absolute values of the numbers being discussed.

**[0144]** The frequency with which LPHDR arithmetic elements may yield only approximations to correct results may vary from implementation to implementation. For example, consider an embodiment in which LPHDR arithmetic elements can perform one or more operations (perhaps including, for example, trigonometric functions), and for each operation the LPHDR elements each accept a set of inputs drawn from a range of valid values, and for each specific set of input values the LPHDR elements each produce one or more output values (for example, simultaneously computing both sin and cos of an input), and the output values produced for a specific set of inputs may be deterministic or non-deterministic. In such an example embodiment, consider further a fraction  $F$  of the valid inputs and a relative error amount  $E$  by which the result calculated by an LPHDR element may differ from the mathematically correct result. In certain embodiments of the present invention, for each LPHDR arithmetic element, for at least one operation that the LPHDR unit is capable of performing, for at least fraction  $F$  of the possible valid inputs to that operation, for at least one output signal produced by that operation, the statistical mean, over repeated execution, of the numerical values represented by that output signal of the LPHDR unit, when executing that operation on each of those respective inputs, differs by at least  $E$  from the result of an exact mathematical calculation of the operation on those same input values, where  $F$  is 1% and  $E$  is 0.05%. In several other example embodiments,  $F$  is not 1% but instead is one of 2%, or 5%, or 10%, or 20%, or 50%. For each of these example embodiments, each with some specific value for  $F$ , there are other example embodiments in which  $E$  is not 0.05% but instead is 0.1%, or 0.2%, or 0.5%, or 1%, or 2%, or 5%, or 10%, or 20%. These varied embodiments are merely examples and do not constitute limitations of the present invention.

**[0145]** For certain devices (such as computers or processors or other devices) embodied according the present invention, the number of LPHDR arithmetic elements in the device (e.g., computer or processor or other device)



exceeds the number, possibly zero, of arithmetic elements in the device which are designed to perform high dynamic range arithmetic of traditional precision (that is, floating point arithmetic with a word length of 32 or more bits). If NL is the total number of LPHDR elements in such a device, and NH is the total number of elements in the device which are designed to perform high dynamic range arithmetic of traditional precision, then NL exceeds  $T(NH)$ , where  $T()$  is some function. Any of a variety of functions may be used as the function  $T()$ . For example, in certain embodiments,  $T(NH)$  may be twenty plus three times NH, and the number of LPHDR arithmetic elements in the device may exceed twenty more than three times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed fifty more than five times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As yet another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed one hundred more than five times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As yet another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed one thousand more than five times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. As yet another example, in certain embodiments, the number of LPHDR arithmetic elements in the device may exceed five thousand more than five times the number of arithmetic elements in the device, if any, designed to perform high dynamic range arithmetic of traditional precision. Certain embodiments of the present invention may be implemented within a single physical device, such as but not limited to a silicon chip or a chip stack or a chip package or a circuit board, and the number NL of LPHDR elements in the physical device and the number NH of elements designed to perform high dynamic range arithmetic of traditional precision in the physical device may be the total counts of the

respective elements within that physical device. Certain embodiments of the present invention may be implemented in a computing system including more than one physical device, such as but not limited to a collection of silicon chips or chip stacks or chip packages or circuit boards coupled to and communicating with each other using any means (such as a bus, switch, any kind of network connection, or other means of communication), and in this case the number NL of LPHDR elements in the computing system and the number NH of elements designed to perform high dynamic range arithmetic of traditional precision in the computing system may be the total counts of the respective elements within all those physical devices jointly.

**[0146]** Certain embodiments of the present invention may constitute, or may be part of, processors, which are devices capable of executing software to perform computations. Such processors may include mechanisms for storing software, for using the software to determine what operations to perform, for performing those operations, for storing numerical data, for modifying data according to the software specified operations, and for communicating with devices connected to the processor. Processors may be reconfigurable devices, such as, without limitation, field programmable arrays. Processors may be co-processors to assist host machines or may be capable of operating independently of an external host. Processors may be formed as a collection of component host processors and co-processors of various types, such as CPUs, GPUs, FPGAs, or other processors or other devices, which in the art may be referred to as a heterogeneous processor design or heterogeneous computing system, some or all of which components might incorporate the same or distinct varieties of embodiments of the present invention.

**[0147]** Embodiments of the present invention may, however, be implemented in devices in addition to or other than processors. For example, a computer including a processor and other components (such as memory coupled to the processor by a data path), wherein the processor includes components for performing LPHDR operations in any of the ways disclosed herein, is an example of an embodiment of the present invention. More generally, any device or

combination of devices, whether or not falling within the meaning of a “processor,” which performs the functions disclosed herein may constitute an example of an embodiment of the present invention.

**[0148]** More generally, any of the techniques described above may be implemented, for example, in hardware, software tangibly stored on a computer-readable medium, firmware, or any combination thereof. The techniques described above may be implemented in one or more computer programs executing on a programmable computer including a processor, a storage medium readable by the processor (including, for example, volatile and non-volatile memory and/or storage elements), at least one input device, and at least one output device. Program code may be applied to input entered using the input device to perform the functions described and to generate output. The output may be provided to one or more output devices.

**[0149]** Each computer program within the scope of the claims below may be implemented in any programming language, such as assembly language, machine language, a high-level procedural programming language, or an object-oriented programming language. The programming language may, for example, be a compiled or interpreted programming language.

**[0150]** Each such computer program may be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a computer processor. Method steps of the invention may be performed by a computer processor executing a program tangibly embodied on a computer-readable medium to perform functions of the invention by operating on input and generating output. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, the processor receives instructions and data from a read-only memory and/or a random access memory. Storage devices suitable for tangibly embodying computer program instructions include, for example, all forms of non-volatile memory, such as semiconductor memory devices, including EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROMs. Any of the foregoing

Attorney Docket No. A0006-1001C2

may be supplemented by, or incorporated in, specially-designed ASICs (application-specific integrated circuits) or FPGAs (Field-Programmable Gate Arrays). A computer can generally also receive programs and data from a storage medium such as an internal disk (not shown) or a removable disk. These elements will also be found in a conventional desktop or workstation computer as well as other computers suitable for executing computer programs implementing the methods described herein, which may be used in conjunction with any digital print engine or marking engine, display monitor, or other raster output device capable of producing color or gray scale pixels on paper, film, display screen, or other output medium.

### **Claims**

1. A device comprising:

at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,

wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/65,000$  through  $65,000$  and for at least  $X=5\%$  of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least  $X\%$  of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least  $Y=.05\%$  from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and

at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit.

2. The method of claim 1, wherein the at least one first computing device comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine.

3. The device of claim 2, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.

4. The device of claim 3, wherein  $X=10\%$ .

5. The device of claim 3, wherein  $Y=.2\%$ .

6. The device of claim 3, wherein  $X=10\%$  and  $Y=.2\%$ .



Attorney Docket No. A0006-1001C2

7. The device of claim 3, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000.

8. The device of claim 3, wherein the first operation is multiplication.

9. A device comprising:

at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,

wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/65,000 through 65,000 and for at least  $X=5\%$  of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least  $X\%$  of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least  $Y=.05\%$  from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and

wherein the at least one first LPHDR execution unit comprises at least one of a chemistry-technology-based execution unit, a biological-technology-based execution unit, a DNA-technology-based execution unit, a nanomechanical-technology-based execution unit, a nanoelectronic-technology-based execution unit, a nanowire-technology-based execution unit, a nanotube-technology-based execution unit, and an optical-technology-based execution unit.

10. The device of claim 9, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.

11. The device of claim 10, wherein  $X=10\%$ .

12. The device of claim 10, wherein  $Y=.2\%$ .

Attorney Docket No. A0006-1001C2

13. The device of claim 10, wherein  $X=10\%$  and  $Y=.2\%$ .

14. The device of claim 10, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000.

15. The device of claim 10, wherein the first operation is multiplication.

16. A device comprising:

a plurality of components comprising:

at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,

wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/65,000 through 65,000 and for at least  $X=5\%$  of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least  $X\%$  of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least  $Y=.05\%$  from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.

17. The device of claim 16, wherein the plurality of components are arranged in a stack.

18. The device of claim 17, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.

19. The device of claim 18, wherein  $X=10\%$ .

20. The device of claim 18, wherein  $Y=.2\%$ .

Attorney Docket No. A0006-1001C2

21. The device of claim 18, wherein  $X=10\%$  and  $Y=.2\%$ .

22. The device of claim 18, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000.

23. The device of claim 18, wherein the first operation is multiplication.

24. The device of claim 16, wherein the plurality of components are bonded.

25. The device of claim 16, wherein the plurality of components are arranged in a stack and bonded.

26. A device comprising a computer processor and a computer-readable memory storing computer program instructions, wherein the computer program instructions are executable by the processor to emulate a second device comprising:

at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value;

wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/65,000 through 65,000 and for at least  $X=5\%$  of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least  $X\%$  of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least  $Y=.05\%$  from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and

wherein the at least one first LPHDR execution unit comprises at least one of a chemistry-technology-based execution unit, a biological-technology-based execution unit, a DNA-technology-based execution unit, a nanomechanical-

Attorney Docket No. A0006-1001C2

technology-based execution unit, a nanoelectronic-technology-based execution unit, a nanowire-technology-based execution unit, a nanotube-technology-based execution unit, and an optical-technology-based execution unit.

27. The device of claim 26, wherein the number of LPHDR execution units in the second device exceeds by at least one hundred the non-negative integer number of execution units in the second device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.

28. The device of claim 27, wherein  $X=10\%$ .

29. The device of claim 27, wherein  $Y=.2\%$ .

30. The device of claim 27, wherein  $X=10\%$  and  $Y=.2\%$ .

31. The device of claim 27, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000.

32. The device of claim 27, wherein the first operation is multiplication.

33. A device comprising a computer processor and a computer-readable memory storing computer program instructions, wherein the computer program instructions are executable by the processor to emulate a second device comprising:

- a plurality of components comprising:

- at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value;

- wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/65,000 through 65,000 and for at least  $X=5\%$  of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least  $X\%$  of the possible valid inputs to the first operation, of the numerical values

Attorney Docket No. A0006-1001C2

represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least  $Y=.05\%$  from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.

34. The device of claim 33, wherein the plurality of components are arranged in a stack.

35. The device of claim 34, wherein the number of LPHDR execution units in the second device exceeds by at least one hundred the non-negative integer number of execution units in the second device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.

36. The device of claim 35, wherein  $X=10\%$ .

37. The device of claim 35, wherein  $Y=.2\%$ .

38. The device of claim 35, wherein  $X=10\%$  and  $Y=.2\%$ .

39. The device of claim 35, wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from  $1/1,000,000$  through  $1,000,000$ .

40. The device of claim 35, wherein the first operation is multiplication.

41. The device of claim 33, wherein the plurality of components are bonded.

42. The device of claim 33, wherein the plurality of components are arranged in a stack and bonded.



**Abstract**

A processor or other device, such as a programmable and/or massively parallel processor or other device, includes processing elements designed to perform arithmetic operations (possibly but not necessarily including, for example, one or more of addition, multiplication, subtraction, and division) on numerical values of low precision but high dynamic range ("LPHDR arithmetic"). Such a processor or other device may, for example, be implemented on a single chip. Whether or not implemented on a single chip, the number of LPHDR arithmetic elements in the processor or other device in certain embodiments of the present invention significantly exceeds (e.g., by at least 20 more than three times) the number of arithmetic elements, if any, in the processor or other device which are designed to perform high dynamic range arithmetic of traditional precision (such as 32 bit or 64 bit floating point arithmetic).

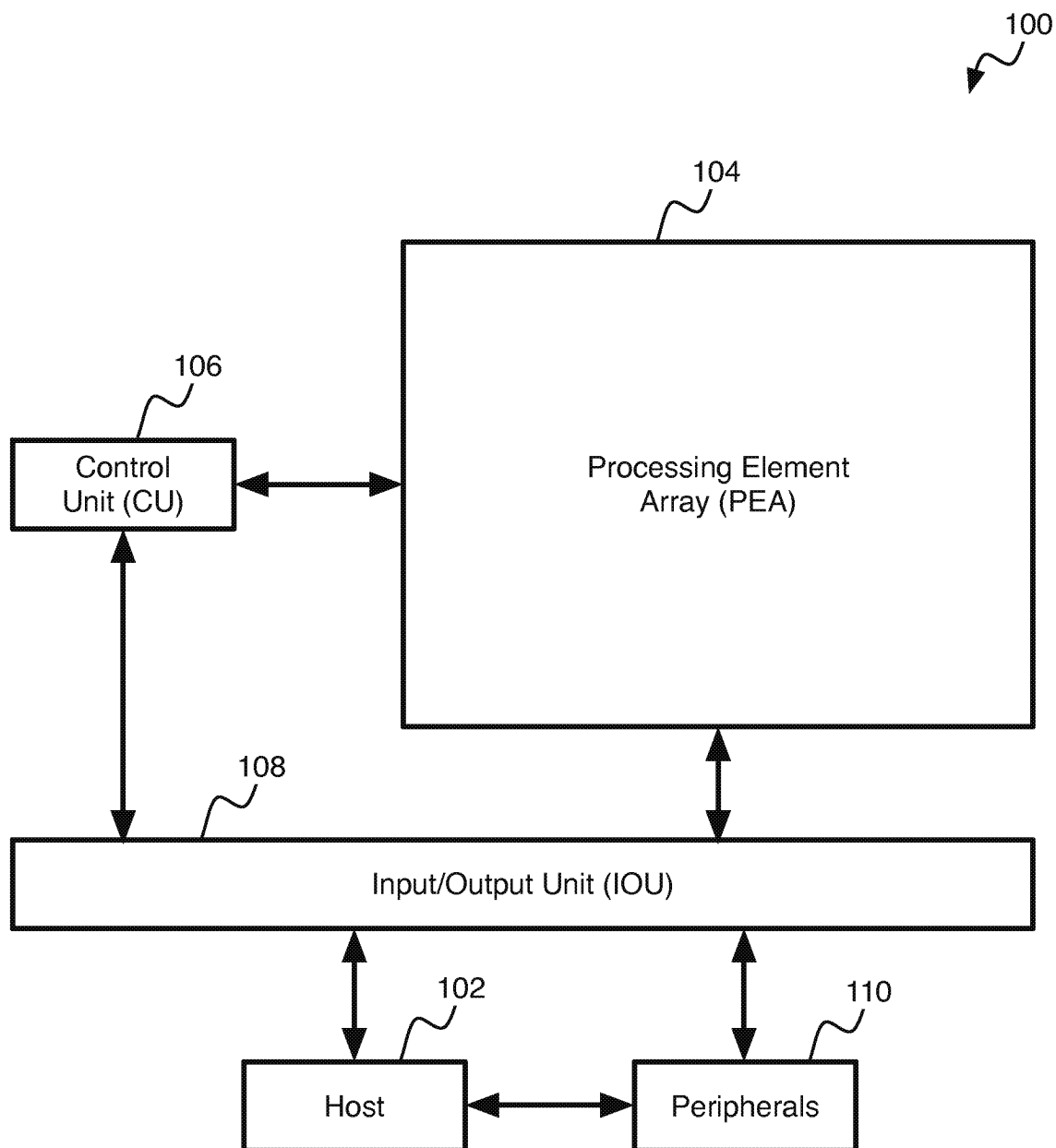


FIG. 1

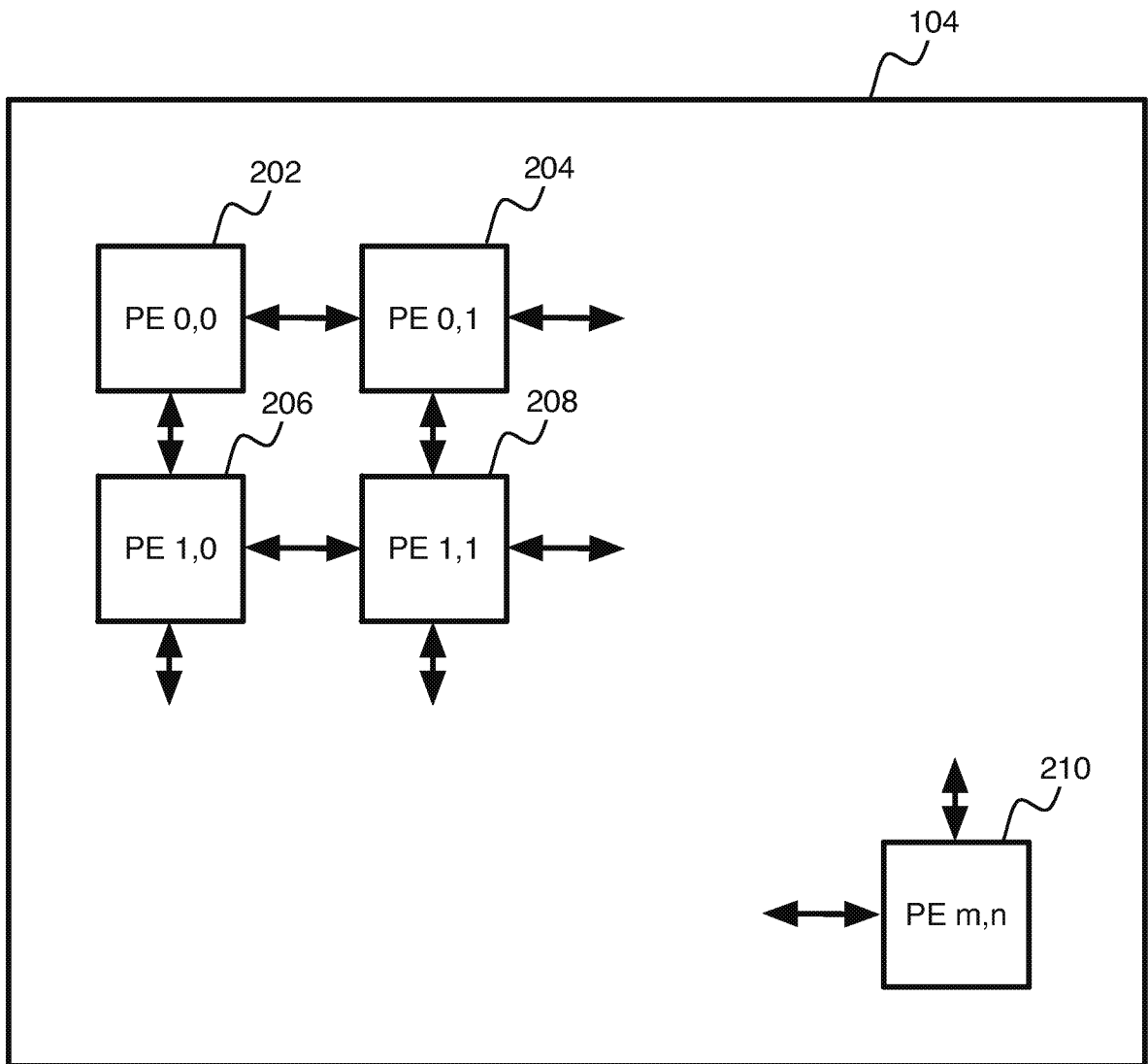


FIG. 2

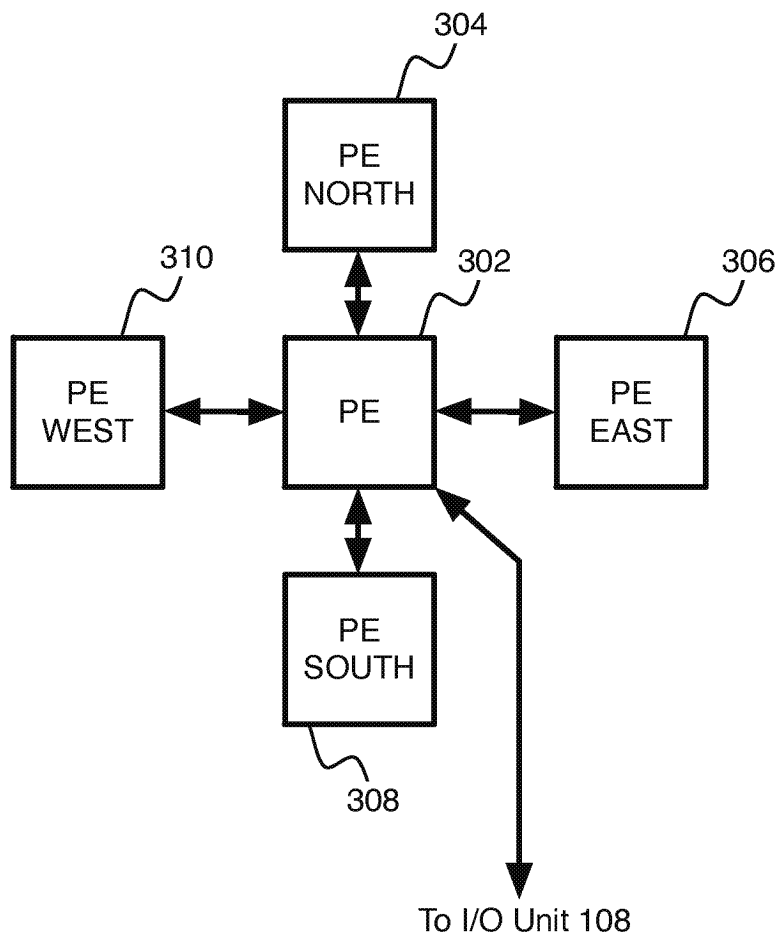


FIG. 3

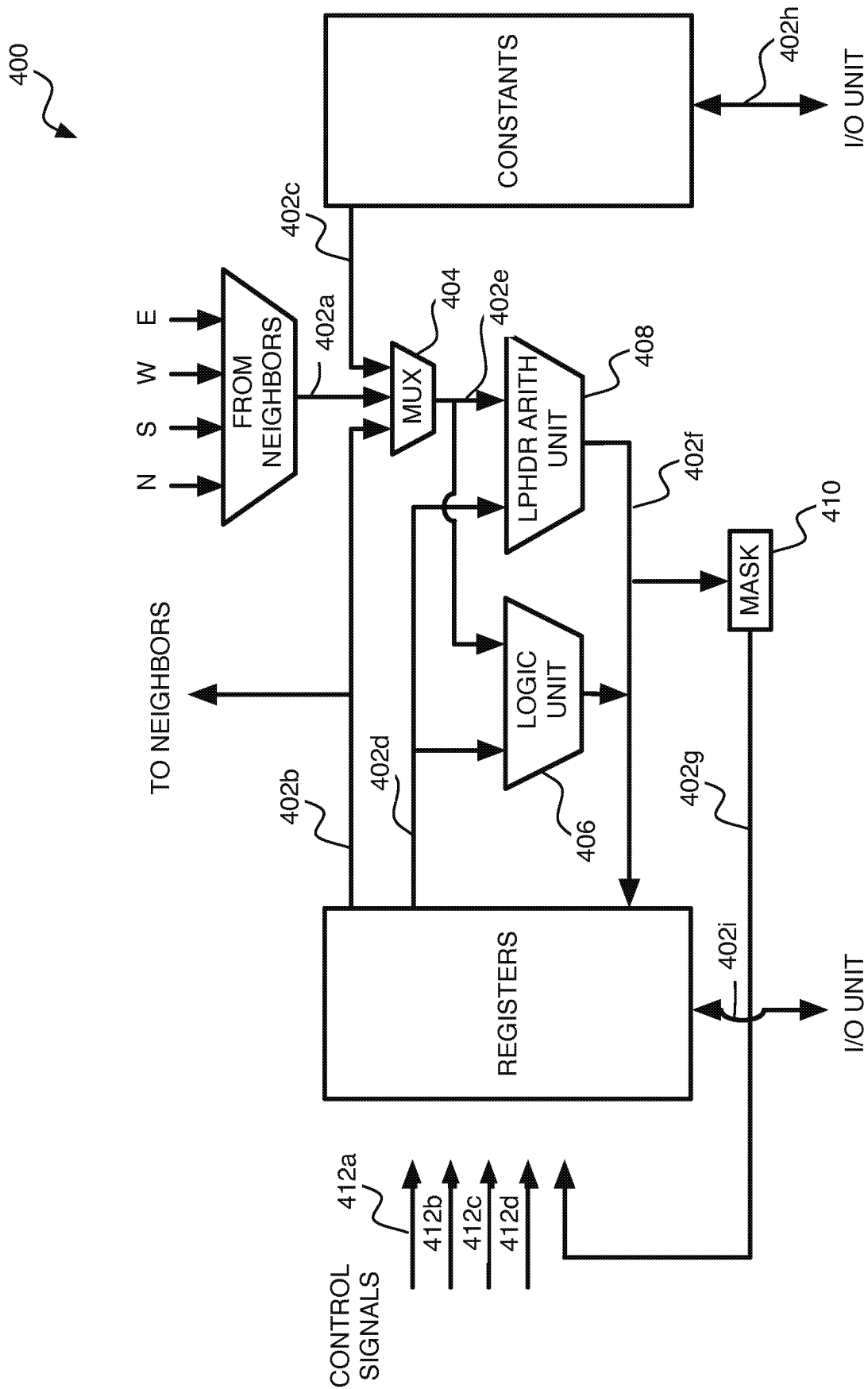


FIG. 4

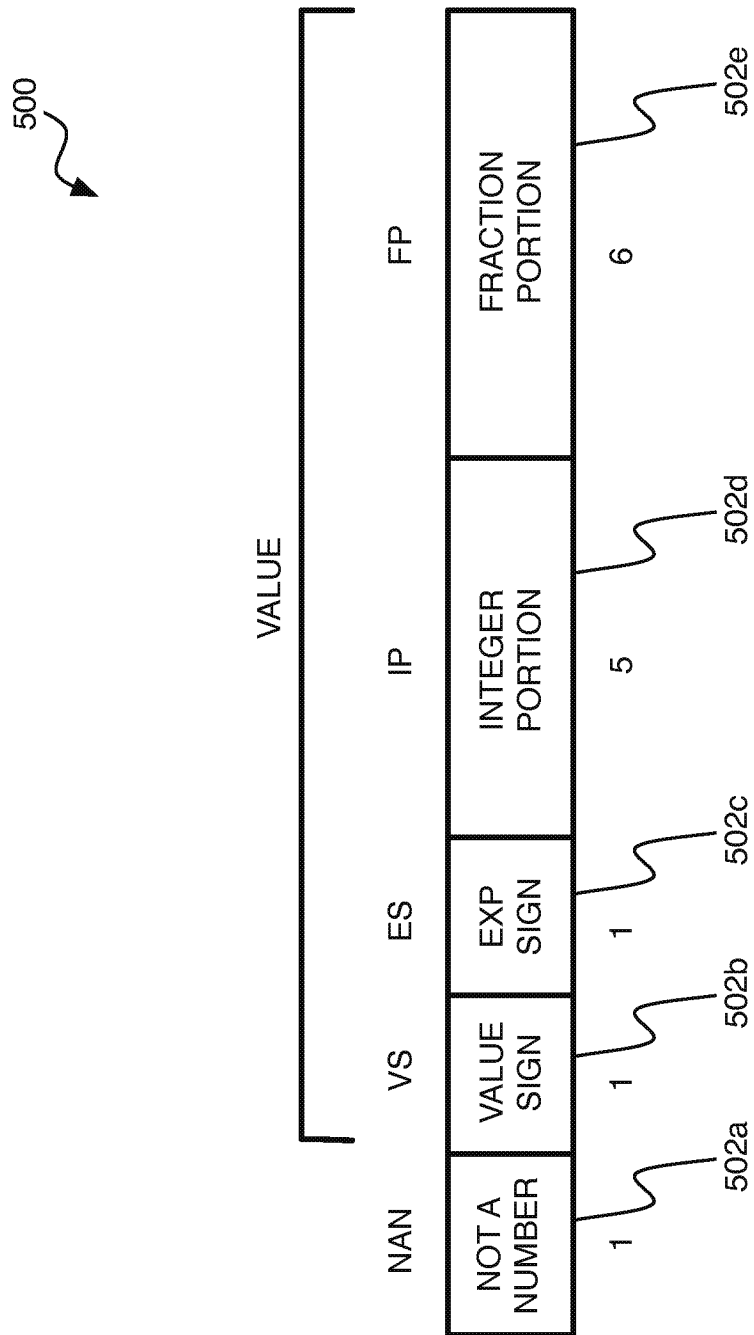


FIG. 5



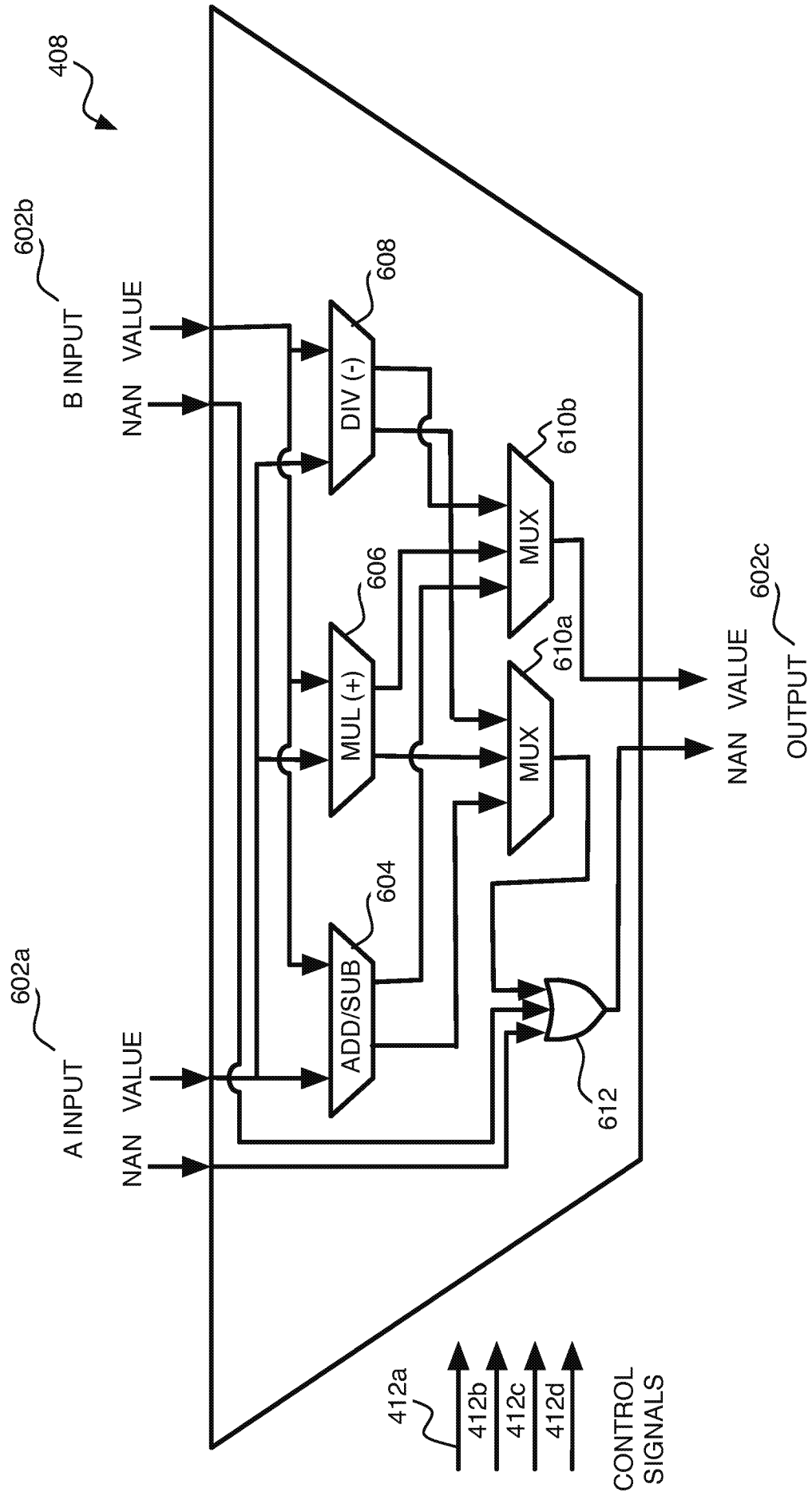


FIG. 6



FIG. 7

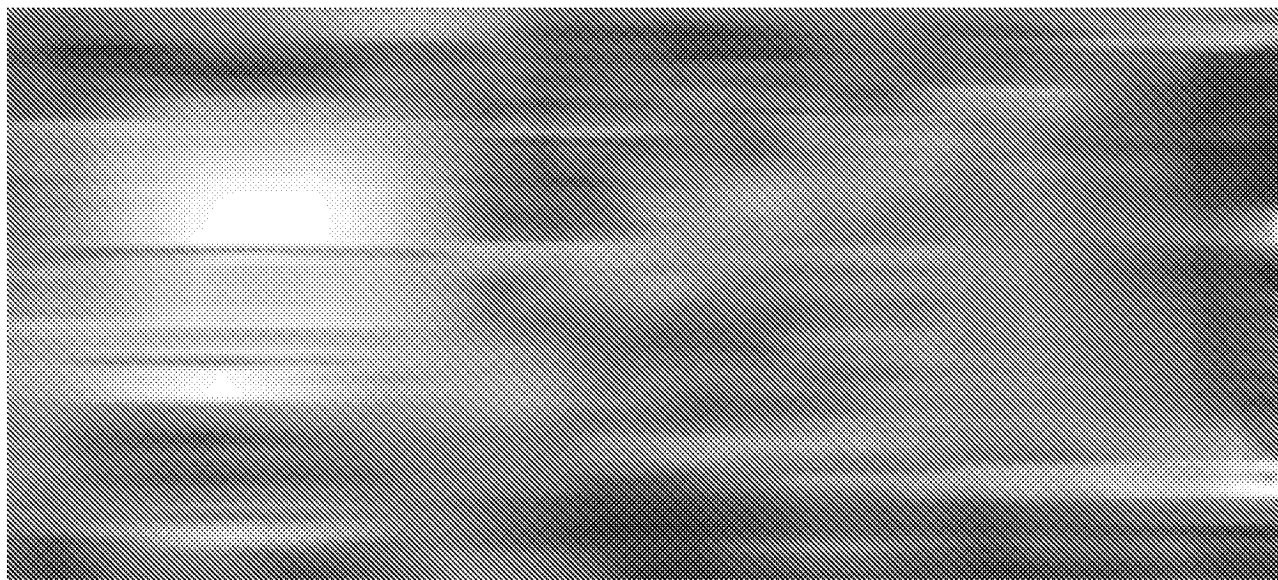


FIG. 8





FIG. 9



FIG. 10





FIG. 11



## Electronic Patent Application Fee Transmittal

<b>Application Number:</b>				
<b>Filing Date:</b>				
<b>Title of Invention:</b>	Processing with Compact Arithmetic Processing Element			
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates			
<b>Filer:</b>	Robert Plotkin/Karen Del Greco			
<b>Attorney Docket Number:</b>	A0006-1001C2			
Filed as Small Entity				
<b>Utility under 35 USC 111(a) Filing Fees</b>				
<b>Description</b>	<b>Fee Code</b>	<b>Quantity</b>	<b>Amount</b>	<b>Sub-Total in USD(\$)</b>
<b>Basic Filing:</b>				
Utility filing Fee (Electronic filing)	4011	1	70	70
Utility Search Fee	2111	1	300	300
Utility Examination Fee	2311	1	360	360
<b>Pages:</b>				
<b>Claims:</b>				
Claims in excess of 20	2202	22	40	880
Independent Claims in Excess of 3	2201	2	210	420
<b>Miscellaneous-Filing:</b>				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
<b>Petition:</b>				
<b>Patent-Appeals-and-Interference:</b>				
<b>Post-Allowance-and-Post-Issuance:</b>				
<b>Extension-of-Time:</b>				
<b>Miscellaneous:</b>				
<b>Total in USD (\$)</b>				<b>2030</b>

**Electronic Acknowledgement Receipt**

<b>EFS ID:</b>	15340277
<b>Application Number:</b>	13849606
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	6059
<b>Title of Invention:</b>	Processing with Compact Arithmetic Processing Element
<b>First Named Inventor/Applicant Name:</b>	Joseph Bates
<b>Customer Number:</b>	24208
<b>Filer:</b>	Robert Plotkin/Karen Del Greco
<b>Filer Authorized By:</b>	Robert Plotkin
<b>Attorney Docket Number:</b>	A0006-1001C2
<b>Receipt Date:</b>	25-MAR-2013
<b>Filing Date:</b>	
<b>Time Stamp:</b>	11:40:40
<b>Application Type:</b>	Utility under 35 USC 111(a)

**Payment information:**

Submitted with Payment	no
------------------------	----

**File Listing:**

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	ADS_A0006-1001C2.pdf	1503510 4377a5d36108babf60f655f283291a3e4faf9c4a	no	6

**Warnings:****Information:**

2	Specification	Specification.pdf	217009 b262df3582210226ed205242567db31302a41f67	no	57
<b>Warnings:</b>					
<b>Information:</b>					
3	Drawings-only black and white line drawings	Drawings.pdf	2313923 12b129b8f06ff7a213a6c43cff5ac7d4d73af3d	no	11
<b>Warnings:</b>					
<b>Information:</b>					
4	Fee Worksheet (SB06)	fee-info.pdf	37731 667d063eb3c3d0a59876b8b994296128b28cba69	no	2
<b>Warnings:</b>					
<b>Information:</b>					
<b>Total Files Size (in bytes):</b>				4072173	
<p><b>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</b></p> <p><b><u>New Applications Under 35 U.S.C. 111</u></b>  <b>If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</b></p> <p><b><u>National Stage of an International Application under 35 U.S.C. 371</u></b>  <b>If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</b></p> <p><b><u>New International Application Filed with the USPTO as a Receiving Office</u></b>  <b>If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</b></p>					

## **SCORE Placeholder Sheet for IFW Content**

Application Number: 13849606

Document Date: 03/25/2013

The presence of this form in the IFW record indicates that the following document type was received in electronic format on the date identified above. This content is stored in the SCORE database.

- Drawings – Other than Black and White Line Drawings

Since this was an electronic submission, there is no physical artifact folder, no artifact folder is recorded in PALM, and no paper documents or physical media exist. The TIFF images in the IFW record were created from the original documents that are stored in SCORE.

To access the documents in the SCORE database, refer to instructions developed by SIRA.

At the time of document entry (noted above):

- Examiners may access SCORE content via the eDAN interface.
- Other USPTO employees can bookmark the current SCORE URL (<http://es/ScoreAccessWeb/>).
- External customers may access SCORE content via the Public and Private PAIR interfaces.